برمجة التحكم المنطقي
P.L.C.

أعطال - صيانة
تمارين

إعداد
ريمون كمال

معهد السالزيان الايطالى (دون بوسكو)

برمجة التحكم المنطقي P.L.C. أعطال - صيانة - تمارين إعداد ريمون كمال

برمجة التحكم المنطقى .P.L.C

الجزء الأول

إعداد

ريمون كمال

معهد السالزيان الإيطالى "**دون بوسكو**"

٢شارع عبد القادر طه - الساحل ت: ٢٤٥٧٩٦٥٠- ٢٤٥٧٦٧٩٤

معهد فنى - معهد صناعى

دورات تدريبية سريعة مركزة

دورات تدريبية تعليمية للمدرسين

المراجع

1. أجهزة التحكم المبرمج وتطبيقاتها العملية.
2. Controllore a logica programmabile P. Bani
3. Siemens Programmable Controller Manual

طبعة جديدة

2011

أسم الكتاب: **برمجة التحكم المنطقى *P.L.C.***

الجزء الأول

طباعة:

رقم الإيداع:

الترقيم الدولى:

شكر و إهداء

أهدى هذا الكتاب إلى أبى وأمي الذين لهم كل الفضل بأن أعمل في هذا المجال وهم الذين شجعوني على عمل هذا الكتاب بكل جهد وإخلاص شاكر الله و إياهم وكل من ساهم في تقديمه.

وأشكر أيضاً كل المعلمين الأفاضل الذين ساعدوا على خروج هذا الكتاب إلى المليء.

☜ **المدير الإيطالي للمعهد:** الأب رينسو ليوناردوسى

☜ **الناظر السابق للمعهد:** الأب بيرناردو أشيربونى

☜ **مدير الدورات التدريبية:** أ. ماجد چورچ

☜ **أستاذ التحكم:** أ. نبيل رزق – أ. وجية جرجس

☜ **أستاذ التكيف والتبريد:** أ. إميل فتح الله

☜ **أستاذ الــ PLC:** أ. ماجد موريس – أ. ماجد عريان – أ. چيوليو جالو – أ. محسن أنتون

أتشرف باستقبال تعليقات السادة القراء بخصوص هذا الكتاب على عنوان البريد الإلكترونى التالى

plcbook@hotmail.com

مقدمة

نظراً للتقدم العلمى السريع المرتبط بالمجال الصناعى وخاصة من الناحية الكهربية أصبح لا غنى عن الربط بين عالم الصناعة وبين التكنولوچيا العصرية ويتمثل هذا الربط بواسطة استخدام أجهزة التحكم المنطقى بمختلف أنوعها والتى تستحق أن تسمى بالأجهزة الذكية نظراً لما تقدمه فى المجال الصناعى من: سهولة فى تصميم البرامج، ومرونة فى أكتشاف الأعطال، ومساعدة فى حل المشاكل، ...الخ

و نظراً لصعوبة ترجمة بعض المصطلحات الخاصة بهذا المجال وخاصة لكى لا تفقد المعنى التقنى أو الفنى لها، تمت كتابتها بلغتها الأصلية لذلك لا تهتم كثيراً عزيزى القارئ بهذه المصطلحات فستكون بسيطة ومفهومة بمجرد ما أن تتعمق بفهم فى هذا المجال.

هكذا أيضاً لا تتعجل عزيزى القارئ فى النظر إلى مواضيع متباعدة خاصة أن كنت بمبتدئ فى هذا المجال وهذا لأنه قد تم شرح المنهج بطريقة متسلسلة ولذلك يفضل للقارئ قراءة المواضيع بالتسلسل التى كتبت به لفهم جميع الأمور دون تخبط.

و خاصاً لفهم التمارين لا يشترط فقط القراءة بترتيب بل يجب أيضاً أن تربط كل شرح و كل رمز بالرسم الموجود ولا تقوم بالقراءة بطريقة عابرة.

تم شرح البرمجة بطريقة عامة دون اللجؤ إلى ماركة بعينها وهذا لكى يخدم كل من يعمل مع وحدات التحكم المنطقى بمختلف أنواعها.

تنقسم معرفة أجهزة التحكم المنطقى إلى أمور عديدة من أهمها:

تصميم برامج – اكتشاف أعطال - حل مشاكل

قد تم التركيز بشكل كبير فى الجزء الأول من هذا الكتاب على معرفة تصميم البرامج بطريقة سلسة وباستخدام أسهل لغات البرمجة.

لذلك أقدم لكم هذا الكتاب لخدمة كل من يدرس أو يعمل فى هذا المجال و أتمنى من الله أن يجد كل من يقرأ هذا الكتاب نفعاً له.

المــؤلــف

الباب الأول

جهاز التحكم المنطقى

- ما هو جهـــــــــــــــــــــــــاز الــ PLC.
- لمـــــــــــــــــــــــــإذا يستخدم جهاز الــ PLC.
- مكـــــــــــــــــــــــــونات جهاز الــ PLC.
- تصنيـــــــــــــــــــــــــف جهاز الــ PLC.
- حمايـــــــــــــــــــــــــات لجهاز الــ PLC.
- توصـــــــــــــــــــــــــيل جهاز الــ PLC.
- لمبـــــــــــــــــــــــــات الإشارة.
- كابل البرمجـــــــــــــــــــــــــة.
- الذاكرة الداخليـــــــــــــــة و الخارجيـــــــــــــة.
- توصيـــــــــــــــل وحدة مدخلات و مخرجات أضافية.
- التحـــــــــــــــــكم بواسطة كمبيوتر أو شاشة.

ما هو الـــ PLC؟

كلمة PLC هى اختصار لكلمة Programmable Logic Control و هى تعنى برمجة التحكم المنطقى.

صنع أول جهاز تحكم مبرمج فى شركة (جينرال موتورز - general motors) عام 1968. وكان الجهاز فى البداية يحل محل الريليهات التقليدية فقط غير أنه لم يكن قادراً على تحقيق متطلبات الشركة المصنعة ولكنه كان فى الحقيقة بداية لجيل جديد فى صناعة الأجهزة القابلة للبرمجة والتى تطورت فيما بعد، وانتشرت بكثرة فى جميع ميادين الصناعة.

وفى الفترة ما بين عامى 1970 و 1974 ونتيجة للتقدم التكنولوچى فى صناعة الميكروبروسيسور أصبحت الأجهزة القابلة للبرمجة أكثر مرونة و ذكاء، وأصبح من السهل على الفنيين و المهندسين الذين ليس لديهم معرفة كبيرة بعلوم الكمبيوتر و الإلكترونيات الرقمية التعامل معها، بل وأصبحت هذه الأجهزة قادرة على القيام بالعمليات الحسابية و المنطقية و أصبح يمكن التحكم بها باستخدام لغات مختلفة أسهل من التى كانت تستخدم فى ما قبل.

أما فى الفترة ما بين عامى 1975 و 1979 حدث تقدم كبير فى صناعة الأجهزة القابلة للبرمجة، وأشتمل هذا التطور على زيادة سعة الذاكرة وعدد المداخل و المخارج الرقمية بل أشتمل هذا التطور أيضاً على زيادة قدرة الميكروبروسيسور فى سرعة تنفيذ البرنامج.

وكذلك أصبح من السهل تخزين أى برنامج فى وحدة ذاكرة خارجية، وأصبح من الممكن تغيير البيانات سابقة التخزين أثناء التشغيل، فأصبح بوسع وحدة البرمجة تغيير قيم المؤقتات الزمنية المبرمجة والعدادات

المبرمجة و نقلات القيم المتغيرة و مفاتيح المقارنة...الخ، بدون إيقاف خطوط الإنتاج الصناعية كما كان فى السابق.

ملاحظة:

فى بعض مجالات الصناعة لا يمكن لوحدة الــ PLC التوقف لتعديل البيانات ولذلك فأنه تم التغلب على هذه المشاكل فيما بعد بواسطة وحدات PLC ذات كفاءة أعلى.

ونتيجة لتطور علوم الأتصالات فى هذه الفترة أصبح من الممكن استخدام مجموعة من أجهزة التحكم المنطقى للعمل سوياً فى شبكة محلية للتحكم فى المصنع، كما لو كانت جهازاً واحداً، و أيضاً من الممكن عمل تقارير وافية عن الإنتاج والصيانة و الأعطال بواسطة الوحدات الخارجية التى توصل إلى جهاز الــ PLC مثل الطابعات أو شاشات التحكم بالمس، وتخدم هذه التقارير إدارات المصانع لتحسين معدل الإنتاج أو تساعد فى الكشف عن الأعطال حيث يمكن طباعة الأعطال التى حدثت فى فترة زمنية معينة.

نظراً للإقبال الشديد فى المجال الصناعى على وحدات البرمجة الذكية PLC تنافست الشركات المصنعة فى تطوير الجهاز وكان نتيجة التطورات الهائلة فى تكنولوچيا صناعة أجهزة التحكم المنطقى ما يلى:

١- أصبحت تكلفة الجهاز منخفضة إلى الحد الذى يسمح باستخدام ه بدلاً من عشرات الريليهات.

٢- أصبح من الممكن استخدام أجهزة التحكم صغيرة الحجم فى التحكم التناظرى analog.

٣- أصبح من الممكن توصيل أجهزة التحكم مع الحساسات الحرارية وأجهزة قياس الانفعال...الخ.

٤- ظهرت أحجام مختلفة من الوحدات المنطقية القابلة للبرمجة فمنها ما يكون عدد مداخله ومخارجه حوالى عشرة فقط، ومنها ما يصل عدد مداخله ومخارجه إلى عدة ألاف.

أدت أيضآ التطورات الهائلة فى أنظمة البرمجة لأجهزة التحكم المبرمج إلى:

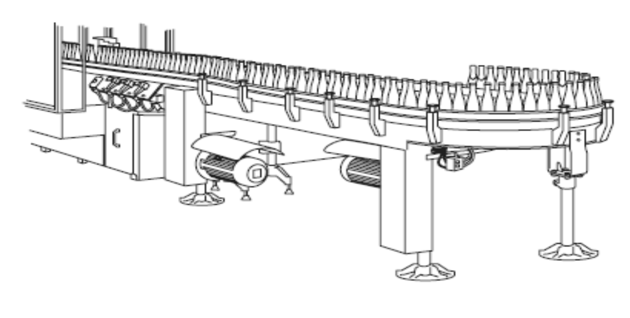
١- استخدام لغات يسهل على من ليس لديه معرفه بعلوم الكمبيوتر استخدامها.

٢- أمكانية تحديد الأعطال وتعديل البيانات الداخلية أثناء تشغيل العملية الصناعية.

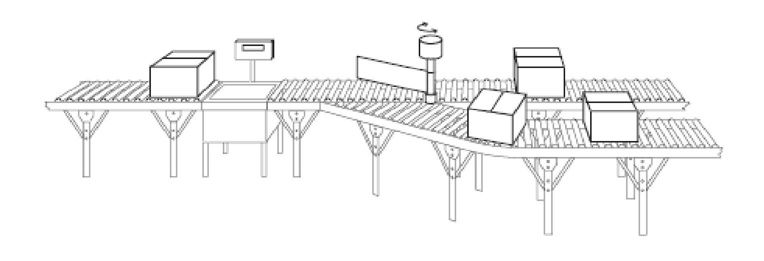
٣- أصبح زمن الاستجابة لأجهزة التحكم المبرمج سريع جداً تصل إلى قراءة البرنامج حوالى 3000 مرة فى الثانية.

٤- أصبحت تستخدم كابلات لنقل المعلومات من و إلى وحدة البرمجة بسرعة فأقة تصل إلى 187,5 Kbps.

٥- أصبح استخدام وحدة الــ PLC فى المجال الصناعى كثيرآ كما فى الشكل (أ) و الشكل (ب).



الشكل (أ)



الشكل (ب)

مميزات أجهزة التحكم المبرمج:

هناك الكثير من المميزات نذكر منها ما يلى :



١- **التحكم المرن:** والمقصود بالتحكم المرن, سهولة تغيير أداء العملية الصناعية لمواكبة أى توسعات وذلك بتعديل برنامج التشغيل

٢- **الصيانة واكتشاف الأعطال:** إن أجهزة التحكم هى عبارة عن أجهزة ألكترونية لذلك فهى لا تحتاج إلى صيانة وهى معدة لإعطاء بيان عن أعطالها سواء كانت أعطال بسيطة أو أعطال فادحة.

٣- **صغر الحجم مع إمكانيات عالية:** إن أحجام أجهزة التحكم المبرمج صغيرة جداً مقارنة بالدوائر الأخرى فى الكنترول، فيمكن القول أن جهاز تحكم مبرمج أبعاده ١٠سم X ١٥سم X ٢٠سم يمكن أن يحل محل مئات الريلاهات، مئات العدادات، مئات المؤقتات الزمنية, بالإضافة لقدرته العالية للقيام بالعمليات الحسابية بل ويحتوى أيضاً على الكثير من العمليات التى ليس لها مقابل فى الكنترول كما سوف نرى فيما بعد.

٤- **خصائصها لا تتوفر فى أجهزة الكمبيوتر المعتادة:** إن أجهزة التحكم المبرمج معدة للعمل فى البيئة الصناعية التى تتميز بأختلاف كبير فى درجات الحرارة والرطوبة ووجود ضوضاء عالية، وكذلك فهى مصممة على أن يقوم بتركيبها وصيانتها وبرمجتها مهندس الموقع مثل المهندسين الكهربائيين الذين ليس لديهم مهارات خاصة بالالكترونيات الرقمية ولا بعلوم الكمبيوتر.

٥- **يمكن أن تعمل داخل شبكة:** يمكن استخدام مجموعة من وحدات البرمجة المنطقية للتحكم فى الماكينات المختلفة المكونة لخطوط الإنتاج، ثم الربط بين وحدات البرمجة المنطقية بواسطة شبكة محلية يتم من خلالها تبادل البيانات اللازمة للتشغيل، يتم التحكم فى كل منهما باستخدام الكمبيوتر، ويتم تبادل البيانات بين الكمبيوتر و وحدة الــ PLC من خلال شبكة الاتصالات المصغرة.

وحدة التحكم المبرمج....... PLC unit:

يحتوى جهاز الـــ PLC على وحدة معالجة مركزية CPU وهى التى تقوم بقراءة البرنامج وتنفيذه, حيث تقوم بقراءة الدخل (input) و تطبيق البرنامج ومن ثم تقوم بتشغيل الخرج (output).



لماذا تستخدم وحدة الـــ PLC:

الـــ PLC هو جهاز ذات تيكنولوچيا عالية فلذلك باستخدام الـــ PLC يمكن عمل الكثير من التمارين المعقدة ولكن بطرق بسيطة جداً مقارنة بالكنترول. بل يمكن أيضاً عمل بعض التمارين التى لا يمكن أن تصمم بالكنترول (control) و بخلاف ذالك يوجد مئات المؤقتات الزمنية (timers) و العدادات (counters) و بعض الأوامر الأخرى التى ليس لها مقابل أو مثيل فى الكنترول.

مكونات وحدة الـــ PLC:

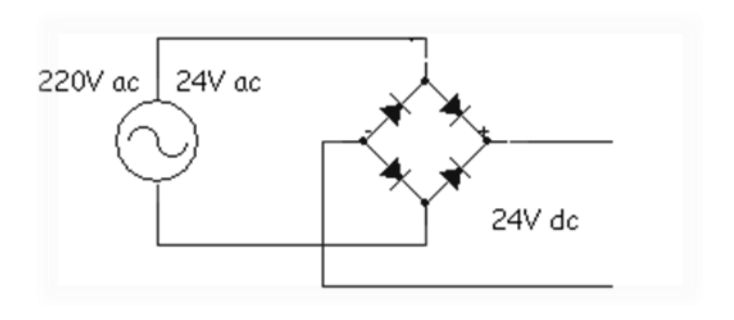
الـــ PLC يتكون من:

١ – وحدة التغذية.........power supply

وهى مكونه من محول خافض للجهد (step down transformer) و دائرة توحيد (rectifier)

المحول الخافض للجهد يقوم بتحويل جهد التيار المتردد إلى جهد متردد أخر أقل قيمة.

دائرة التوحيد تقوم بتحويل التيار المتردد إلى تيار مستمر دون تغير قيمة الجهد.



ملاحظة:

أ – استخدام وحدة التغذية (power supply) يعتمد على نوع الـــ PLC: يوجد بعض الأنواع التى تعمل بـــ 220V AC ولذلك فى هذه الحالة يتم التوصيل بمصدر الكهرباء مباشراً ولا يستخدم الـــ (power supply).

ب –يحتوى جهاز الـــ PLC من الداخل على دائرة توحيد rectifier لتحسين التيار المستمر قدر الأمكان.

ت -في بعض الحالات قد نحتاج أثناء التوصيل إلى استخدام أكثر من وحدة تغذية فى نفس الوقت (power supply) و فى هذه الحالة يفضل أن يتم توصيل الطرف السالب لكل وحدات التغذية المستخدمة معاً لضمان تساوى balance الجهد الخارج من وحدات التغذية.

٢- وحدة المعالجة المركزية..... C.P.U. "Central Processing Unit"

وهى تعتبر العقل المفكر لجهاز الـــ PLC وهو الذى يقراء البرنامج ويقوم بالعمليات الحسابية بطريقة فائقة السرعة بحيث يقوم بتشغيل أو فصل الخرج فى الوقت المناسب.

لكل وحدة معالجة مواصفات خاصة تؤثر على سرعتها فى تنفيذ العمليات, فمثلاً:



- المعالج رقم CPU 313: يعمل على قراءة برنامج بحجم 12KB خلال 0.6ms
- المعالج رقم CPU 314: يعمل على قراءة برنامج بحجم 24KB خلال 0.3ms

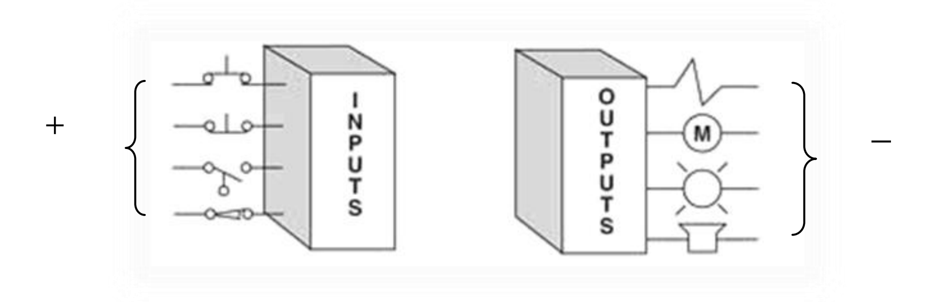
ملاحظة:

- يحتاج المعالج دائماً إلى ذاكرة memory لأحتواء البرنامج بينما يقوم هو بالقراءة والتنفيذ.
- الذاكرة الموجودة داخل الـــ PLC دائماً تكون من النوع المتطاير volatile memory وسوف نتكلم عنها بالتفصيل.

٣- وحدة الدخل و الخرج........input and output modules.

بالنسبة إلى الدخل فهو المكان الذى يتم فيه توصيل طرف واحد فقط من طرفى المفاتيح بحيث يقوم بإرسال الإشارة إلى الــ PLC وهذا يطبق على المفاتيح ذات الإشارة الرقمية بجميع أنوعها.

بالنسبة إلى الخرج فهو المكان الذى يتم فيه توصيل طرف واحد فقط من طرفى الحمل بحيث يقوم باستقبال الإشارة **من** الــ PLC وهذا يطبق على الأحمال ذات الإشارة الرقمية بجميع أنوعها.



ملاحظة:

- يتم توصيل الطرف الأخر من المفاتيح بالتيار الكهربى مع مراعاة قيمة الڤولت المكتوب على جهاز الــ PLC.
- يتم توصيل الطرف الأخر من الأحمال بالسالب مع مراعاة أن جميع أجهزة الــ PLC تعطى إشارة موجبة فى الخرج ولذالك يتم توصيل الطرف الأخر بالسالب.

تنقسم المدخلات إلى مداخل ذات إشارة رقمية و أخرى ذات أشاره تناظرية و تنقسم المخرجات أيضاً إلى مخارج ذات إشارة رقمية و أخرى ذات أشاره تناظرية.

أنواع المداخل والمخارج والفرق بينهما

الإشارة الرقمية:

- **مداخل رقمية:**

توصل المداخل الرقمية بوحدات المداخل الرقمية الخاصة بأجهزة التحكم المنطقية وتقوم وحدة الدخل الرقمية بالتحكم فى أشارات المداخل تبعاً لظروف التشغيل (وعادة فإن قيمة الكهربية يكون لها قيمتين فقط: واحد أو صفر، وتتغير القيمة الكهربية من و إلى هاتين القيمتين حسب حالة المفتاح.

- **مخارج رقمية:**

توصل المخارج الرقمية بوحدات الخرج الرقمية الخاصة بأجهزة التحكم المنطقية وتقوم وحدة الخرج الرقمية بالتحكم بالمخارج، وجميع هذه المخارج لها حالتين فقط: حالة تشغيل، وحالة توقف.

الإشارة التناظرية.

- **مداخل تناظرية:**

توصل أجهزة المداخل التناظرية بوحدات المداخل التناظرية الخاصة بأجهزة التحكم المنطقية وتقوم وحدة الدخل التناظرية بتحويل أى كمية مطلوب قياسها إلى كمية كهربية مثل الجهد والتيار (وعادة فإن القيمة الكهربية يكون لها قيمتين: قيمة عظمى وقيمة صغرى، وتتغير القيمة الكهربية بين هاتين القيمتين.

- **مخارج تناظرية:**

توصل أجهزة المخارج التناظرية بوحدات الخرج التناظرية الخاصة بأجهزة التحكم المنطقية وتقوم وحدة المخارج التناظرية بالتعامل مع القيم المتغيرة الناتجة عن وحدات الخرج الخاصة بوحدة التحكم المنطقى وجميع هذه الأجهزة لها أكثر من حاله: وعادة فإن حالة الخرج تتغير حسب القيمة الكهربية للخرج التناظرى و تتغير القيمة الكهربية بين قيمة عظمى وقيمة صغرى.

أمثلة لمداخل ذات إشارة رقمية.

١– الضواغط اليدوية **Push Buttons**:

وهذه الأجهزة ينعكس حالة ريش تلامسها أى تصبح الريشة المفتوحة طبيعياً مغلقة، والعكس صحيح، وذلك عند الضغط على رؤوسها.



٢– مفاتيح نهاية المشوار **Limit Switches** :

وهذه الأجهزة يتغير حالة تلامسها عند دفع عنصر الفعل لها بكامة متحركة. توجد أنواع مختلفة لمفاتيح نهاية المشوار الميكانيكية.



٣– المفاتيح التقاربية **Proximity Switches** :

وهذه الأجهزة ينعكس وضع ريشة تلامسها عند اقتراب جسم غريب منها لمسافة معينة تعتمد على مدى تشغيل المفتاح التقاربى





٤– مفاتيح الخلايا الضوئية **Photocell Switches** :

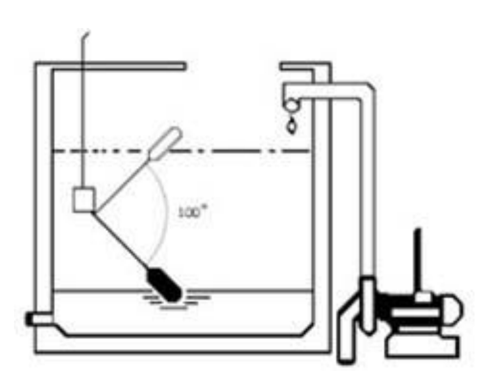
وينعكس حالة ريش تلامس هذه الأجهزة عند مرور جسم غريب ليقطع الشعاع المنبعث من وحدة الإرسال للخلية الضوئية, أى أنه عند مرور أى جسم غريب بين المستقبل والمرسل للخلايا الضوئية تتغير نقط التلامس الخاصة بالخلايا الضوئية.





٥– مفاتيح العوامات Float Switches :

وتستخدم هذه لمفاتيح لتتبع مستوى السوائل فى الخزانات حيث ينعكس حالة ريش هذه المفاتيح عند وصول مستوى السائل فى الخزانات إلى مستواها، فيستخدم فى تتبع مستوى السوائل فى الخزانات.



٦– مفاتيح الضغط Pressure Switches:

وهى أجهزة ينعكس حالة ريش تلامسها عند وصول الضغط فى الأنابيب والخزانات إلى الضغط المضبط مسبقاً، وتستخدم هذه المفاتيح لمتابعة ضغوط السوائل والغازات.



٧- مفاتيح درجة الحرارة **Thermo states** :

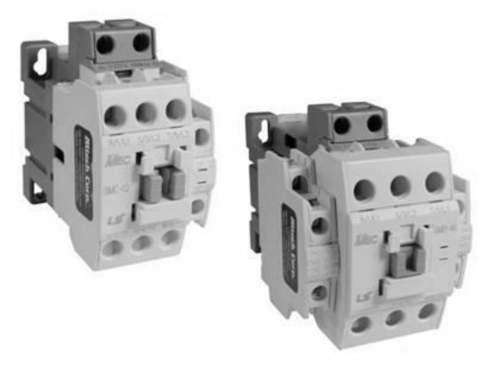
وهى أجهزة ينعكس حالة ريش تلامسها عند ارتفاع درجة حرارة الوسط المحيط لعنصر إحساسها وصولاً لدرجة الحرارة المعايرة عليها هذه المفاتيح.



أمثلة لمخرجات ذات إشارة رقمية.

١- الكونتاكتورات **Contactors** :

ويقوم الكونتاكتور بتوصيل التيار الكهربى إلى الأحمال عند وصول جهد كهربى إلى ملفه والعكس صحيح. يتكون الكونتاكتور من ملف كهربى (البوبينة) وقلب مغناطيسى له شق ثابت، وآخر متحرك يحمل ريش تلامس رئيسية، وعند وصول جهد كهربى على أطراف ملف الكونتاكتور ينجذب الشق المتحرك للقلب تجاه الشق الثابت، فتنعكس ريش الكونتاكتور.



٢- الريليهات الأستاتيكية Static relays :

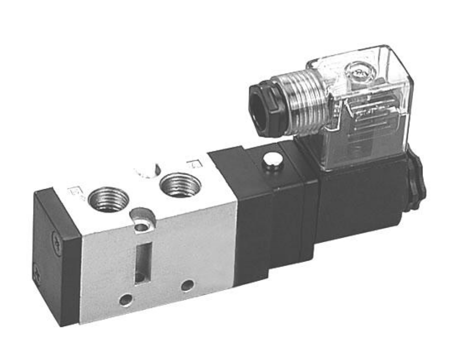
وتقوم بتوصل وفصل التيار الكهربى عن الأحمال - تماماً مثل الكونتاكتورات - ويفضل استخدام ها بدلاً من الكونتاكتورات عند زيادة عدد مرات التوصيل والفصل فى الدقيقة.



٣- المحابس الكهربية Solenoid Valves :

تقوم بفتح أو غلق مسارات مرور السوائل فى الأنابيب وتتكون من: ملف كهربى، وقلب مغناطيسى ثابت وقلب مغناطيسى متحرك، يقوم بفتح أو غلق المحبس وعند وصول تيار كهربى لملف المحبس الكهربى يتحرك الجزء المتحرك للقلب المغناطيسى فيفتح مسار مرور السوائل وهكذا ...





٤– لمبات البيان **Indication Lamps** :

وهى توجد فى غرف التحكم لمساعدة المشغلين على فهم أداء العمليات الصناعية و توضيح الأعطال.



٥– الأبواق **Horns** :

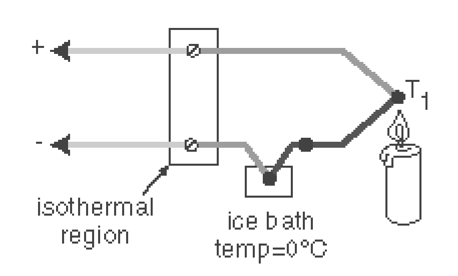
وهى تصدر أصوات عالية عند حدوث أمر غير عادى فى العملية الصناعية لتنبيه المشغلين، وتعمل عند وصول تيار كهربى لملفها, حيث أنها تعتمد على تدفق الهواء المضغوط داخل زور البوق.

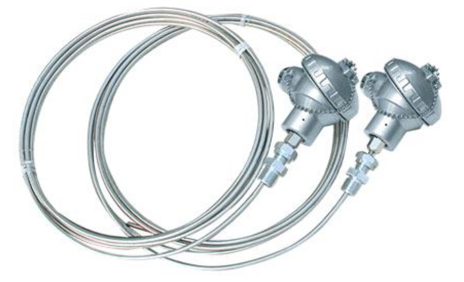


أمثلة لمداخل ذات إشارة تناظرية.

١- الازدواج الحرارى Thermo Couple :

يقوم الازدواج الحرارى بتحويل درجة الحرارة إلى إشارة جهد ويتكون الازدواج الحرارى من معدنين مختلفين (أ - ب) متصلين معاً لتكوين مجس القياس، وتعتمد قيمة الجهد المتولد على درجة حرارة الوصلة, و لذلك فأن الجهد على أطراف الأذدواج الحرارى يتناسب طردياً مع درجة الحرارة.





٢- مولد التاكو Tacho generator :

وهذا المولد يثبت على أعمدة المحركات المطلوب قياس سرعتها، وخرج مولد التاكو خطى،بمعنى أن جهد أطرافه يتناسب طردياً مع السرعة، فإذا كانت نسبة التحويل المولد التاكو هى واحد فولت لكل **٣٠٠** لفة فى الدقيقة الواحدة، فمثلاً إذا كان الجهد على طرف المولد هو ٥ فولت، يعنى هذا أن سرعة المحرك هى ١٥٠٠ لفة فى الدقيقة الواحدة و هكذا ...



أمثلة لمخارج ذات إشارة تناظرية.

١- الدرايفر Driver :

يتم استخدام الدرايفر كمثال للخرج التناظرى حيث يتم استقبال قيمة كهربية (mv – ma – v – Ω) من وحدة البرمجة المنطقية وبواسطة جهاز الدرايفر يتم التحكم فى سرعة الموتور عن طريق تغيير قيمة التردد.



٢- محرك الدامبر Damper motor:

ينم استخدام المحرك الدامبر كمثال للخرج التناظرى حيث يتم استقبال قيمة كهربية (mv – ma – v – Ω) من وحدة البرمجة المنطقية وبواسطة المحرك الدامبر يتم التحكم فى درجة التبريد الخاصة بأجهزة التكييف المركزية حيث يقو بفتح بوابة مرور الهواء البارد حسب الاحتياج.



٤– الذاكرة..........memory

الذاكرة داخل الــ PLC مهمة جداً لأنها تحتوى على البرنامج الذى يقوم الــ CPU بقراءته ولهذا من الممكن فصل الكمبيوتر عن الــ PLC بعد تحميل البرنامج على الذاكرة.



ملاحظة:

- إذا كانت الذاكرة التى تحتوى على البرنامج من النوع المتطاير...volatile memory كما سيوضح فيما بعد فيجب أن يكون جهاز الــ PLC موصل بمصدر مستمر للتيار مثل (battery, plug) لحفظ البرنامج.

- إذا كانت الذاكرة التى تحتوى على البرنامج من النوع غير المتطاير... permanent memory فليس من الضرورى أن يكون جهاز الــ PLC موصل بمصدر مستمر للتيار.

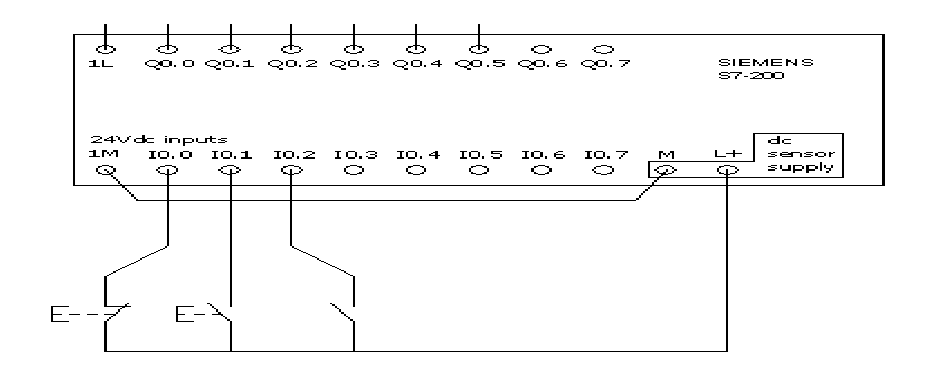
- لا يمكن تخزين أكثر من برنامج على الذاكرة فى نفس الوقت.

تصنيف وحدة الـــ PLC

١– الترانزستور الثنائي القطب...bipolar junction transistor:

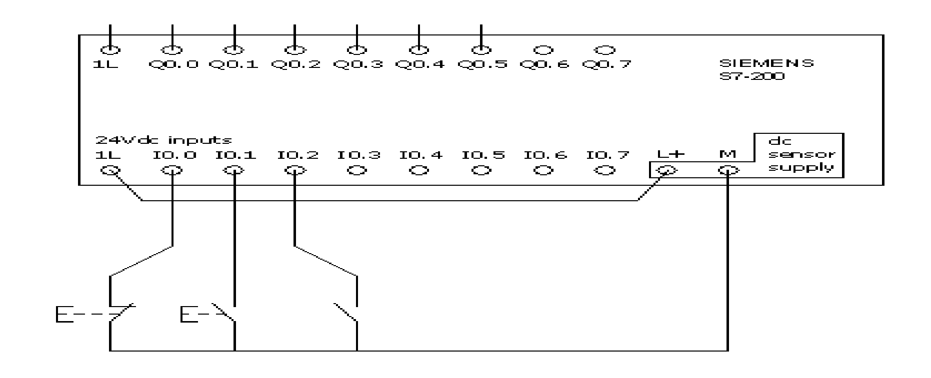
- PNP(Positive Negative Positive):

إذا كان الـــ PLC من النوع PNP يتم تغذية جميع المفاتيح بإشارة موجبة بينما يتم توصيل السالب على وحدة الدخل حيث يوجد حرف M كما هو موضح بالرسم.



- NPN(Negative Positive Negative):

إذا كان الـــ PLC من النوع NPN يتم تغزيه جميع المفاتيح بإشارة سالبة بينما يتم توصيل الموجب على وحدة الدخل حيث يوجد حرف L+ كما هو موضح بالرسم.



ملاحظة:

أ - يفضل أن يكون الــ PLC من نوع NPN وليس PNP لأن فى حالة استخدام PLC (PNP) يكون الطرف الموجب متصل بالمفاتيح لذلك إذا قام العامل بلمس الطرف الموجب بالخطأ بينما تلمس قدميه الأرض سوف يصاب العامل بصدمة كهربية, أما إذا كان الــ PLC (NPN) يكون الطرف السالب متصل بالمفاتيح لذالك إذا قام العامل بلمس الطرف السالب بالخطأ و تلمس قدميه الأرض لن يصاب العامل بصدمة كهربية قط لأن فرق الجهد بين السالب و الأرض يساوى صفر.

ب -نوع الــ PLC سواء كان PNP أو NPN فهذا يشير فقط إلى طريقة توصيل الدخل input وليس له علاقة بطريقة توصيل الخرج output لأن جهاز الــ PLC يعطى إشارة للخرج موجبة دائماً ,كما ذكرت سابقاً, بغض النظر عن نوع جهاز الــ PLC .

ت -حرف الــ L المكتوب على جهاز الــ PLC يعنى مكان توصيل الطرف الموجب بينما حرف الــ M يعنى مكان توصيل الطرف السالب وهذا أن دل على شيء فأنة يدل على أن هذا النوع من أجهزة الــ PLC تعمل بالتيار المستمر.

ث -يعتمد أيضاً استخدام نوع الــ PLC حسب النوع المتوافر فى الأسواق ولذالك فأنها تتغير من قارة إلى قارة أو من بلد لأخرى.

٢- الإشارة الرقمية و الإشارة التناظرية.........Digital & Analog

أ –الإشارة الرقمية.....Digital:

المقصود بالإشارة الرقمية digital هى أى إشارة رقمية لها حالتين فقط, أما أن تساوى الإشارة واحد أما أنها تساوى صفر.

فمثلاً إذا تكلمنا عن digital input فالحالتين هم:

أن كان المفتاح مغلق (YES – TRUE – ON) ويرمز لها بواحد (١).

أن كان المفتاح مفتوح (NO – FALSE – OFF) ويرمز لها بصفر (٠).

وبالمثل إذا تكلمنا عن digital output فالحالتين هم:

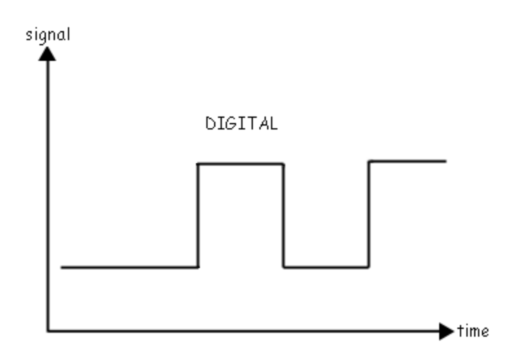
أن كان الخرج يعمل (YES – TRUE – ON) ويرمز لها بواحد (١).

أن كان الخرج لا يعمل (NO – FALSE – OFF) ويرمز لها بصفر (٠).



- مثال للمداخل الرقمية: (مفتاح عادى – مفتاح جرس – مفتاح نهاية المشوار).
- مثال للمخارج الرقمية: (لمبة – جرس – موتور – مضخة).

شكل الإشارة الرقمية:



ب -الإشارة التناظرية.....Analog:

المقصود بالإشارة التناظرية Analog هى أى إشارة لها أكثر من حالتين أى أن الإشارة لها قيم متغيرة بخلاف الصفر.

فمثلاً إذا تكلمنا عن Analog input:

أن كانت توجد إشارة, فقد تكون: (١-٢-٣-٤-......-٣٢٧٦٧).

أن كانت لا توجد إشارة: (صفر).

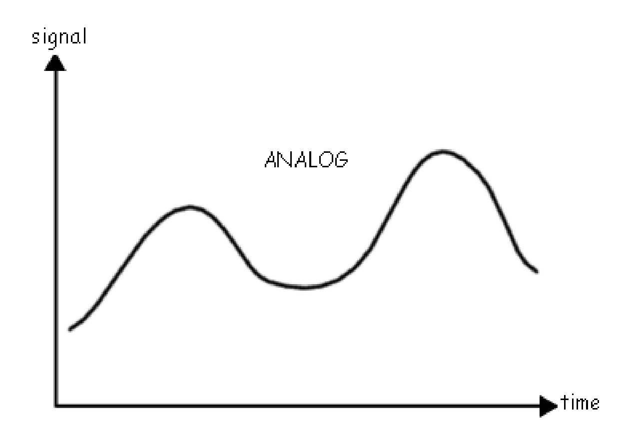
وبالمثل إذا تكلمنا عن Analog output:

أن كانت توجد إشارة, فقد تكون: (١-٢-٣-٤-......-٣٢٧٦٧).

أن كانت لا توجد إشارة: (صفر).



- مثال للمداخل التناظرية: (ترمومتر– المقاومة المتغيرة - إنكودر).
- مثال للمخارج الرقمية: (سخان – الفولتميتر).



ملاحظة:

- رقم ٣٢٧٦٧ هو رقم ناتج عن مساحة الـــ word.

٣- أنواع المخرجات: (ريليه أو ترانزستور) ... Transistor or Relay

النوع الأول: ترانزستور.....Transistor

إذا كان نوع الـــ PLC هو output transistor فهذا لا يجعل طريقة التوصيل تختلف ولكن له بعض المميزات والعيوب مقارنتاً بالنوع الأخر (output relay):

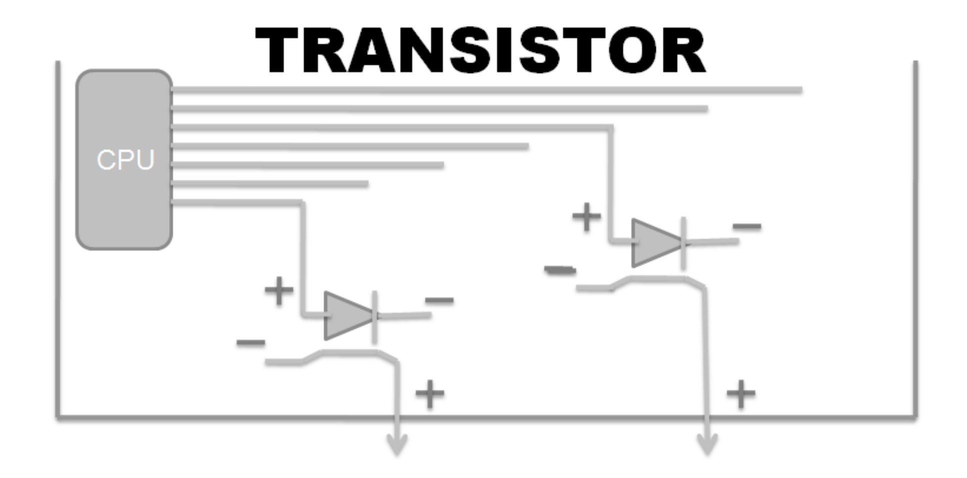
المميزات:

١- يمكنه إرسال إشارات سريعة فى وقت قصير حيث أنة قد يرسل أكثر من ألف إشارة فى الثانية تقريباً.

العيوب:

١- القولت الخاص بالـــ output ثابت وقيمته تساوى 24V DC.

٢- لا يتحمل تيار أكثر من 0.36A – 0.5A.



النوع الثانى: ريليه.......Relay

إذا كان نوع الـــ PLC هو output relay فهذا لا يجعل طريقة التوصيل تختلف ولكن له بعض المميزات والعيوب مقارنتاً بالنوع الأخر (output transistor):

المميزات:

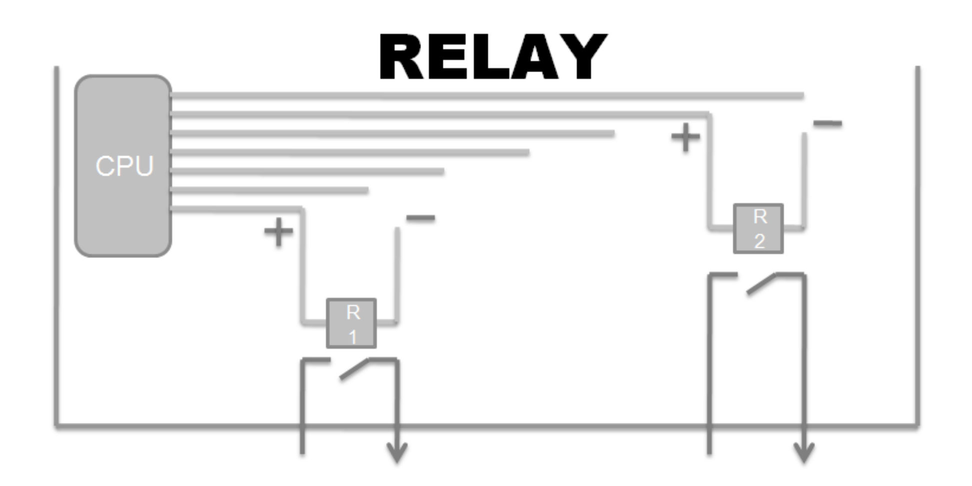
١- الفولت الخاص بالـــ output غير محدد بل من الممكن توصيل أى قيمة ضمن الحد المسموح به.

مثلاً: 12V DC , 110VAC , 220V AC, 24V DC.

٢- يتحمل تيار يصل إلى 2A - 2.5A.

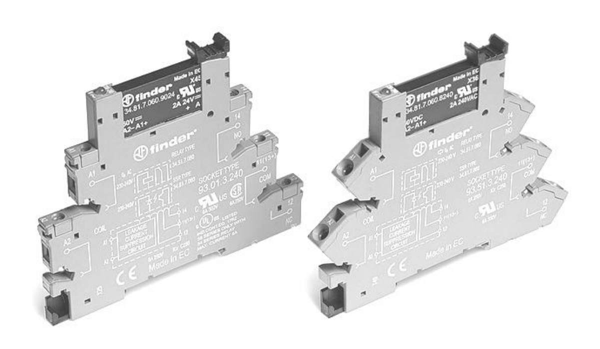
العيوب:

١- لا يمكنه إرسال إشارات سريعة فى وقت قصير مثل الترانزستور.



ملاحظة:

١ – لحل العيوب الموجودة بالنوعين, سواء فى الـــ **output transistor** أو **output relay** يتم استخدام ريليه ميكانيكى **relay interface**.



الريليه الميكانيكى......Relay interface

الريليه الميكانيكى يعمل كوسيط بين الـــ PLC و الحمل حيث يقوم الـــ PLC بتشغيل الريليه بينما يقوم بعد ذلك الريليه بتشغيل الحمل المراد تشغيله بواسطة استخدام نقاط المساعدة.

يتم شراء الريليه على أساس مواصفات تخص الملف coil و مواصفات أخرى تخص النقاط contact.

أولاً: مواصفات الملف...Coil:

١ – أن يكون الجهد الخاص بملف الريليه يعمل بنفس قيمة الجهد الخارج من وحدة الـــ PLC.

٢ – أن يكون التيار المسحوب من ملف الريليه ضمن الحد المسموح به لكى لا يضر بوحدة الـــ PLC.

ثانياً: مواصفات نقاط التلامس...Contact:

١ – أن تتحمل نقاط الريليه الجهد الخاص بالحمل الذى سيعمل بواسطة النقاط المساعدة.

٢ – أن تتحمل نقاط الريليه قيمة التيار المسحوب من الحمل الذى سيعمل بواسطة هذا الريليه.

لماذا يستخدم الريليه؟

١– للتمكن من تشغيل الأحمال التى تعمل بڤولت مختلف عن الڤولت الخارج من جهاز الــ PLC.

٢– للتمكن من تشغيل الأحمال التى تسحب تيار بقيمة أكبر من التى يتحملها جهاز الــ PLC.

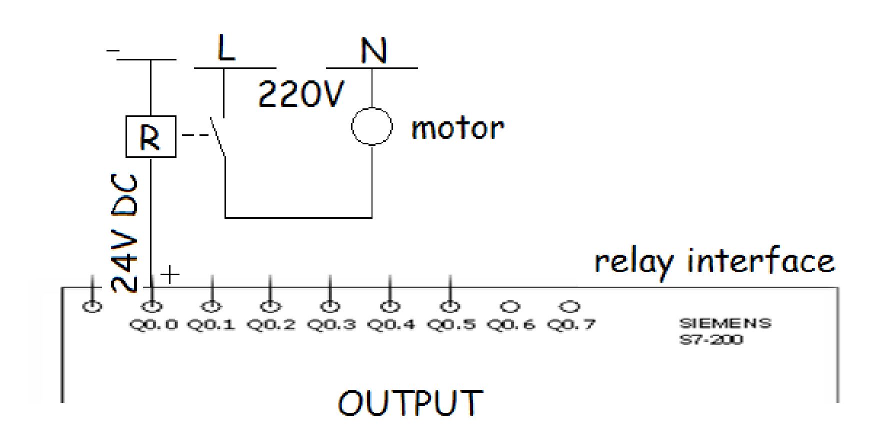
٣– لحماية جهاز الــ PLC من التيار الزائد الذى قد يسحبه الحمل فى أى وقت.

متى يستخدم الريليه؟

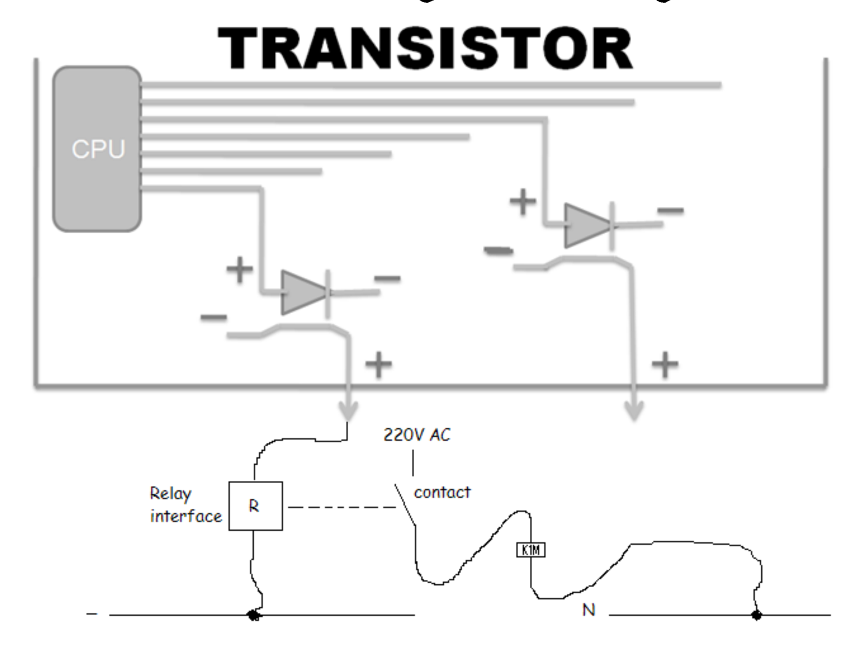
١– إذا كانت الأحمال المستخدمة تختلف فى التيار أو فى الجهد مع جهاز الــ PLC.

٢– وأيضا إذا كان الحمل يتوافق مع الــ PLC من حيث التيار و الجهد ولكن من الممكن إذا كان هذا الحمل محرك أن يسحب تيار زائد لزيادة قوة العزم مثلاً.

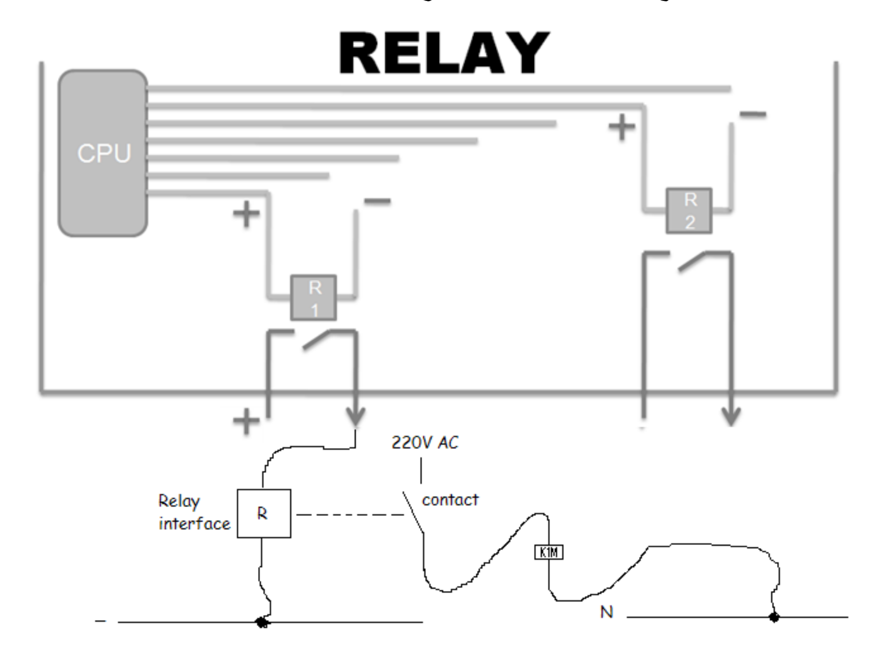
طريقة توصيل الريليه



- طريقة توصيل أحمال مع وحدة PLC من نوع الـــ output transistor:



- طريقة توصيل أحمال مع وحدة PLC من نوع الـــ output relay:



ملاحظة:

١- فى حالة استخدام الريليه يصبح من الممكن تغذية أى حمل يعمل بأى جهد ويسحب أى تيار فى حدود نوع الريليه المستخدم.

٢- إذا سحب الموتور تيار زائد فليس هناك أى خطر على الـــ PLC لأنه سوف يتأثر الريليه وليس الـــ PLC.

٣- أختار الريليه بحيث أن يعمل بنفس الڤولت الخارج من الـــ PLC بينما أختار نقاط الريليه بحيث تتحمل التيار المسحوب من الحمل.

٤- من المحتمل أن الحمل يسحب تيار زائد من نقطة الريليه و لكن ليس من الممكن أن يسحب الريليه تيار زائد من الـــ PLC لأن الريليه ليس إلا ملف solenoid.

٥- قد يجد البعض مشكلة فى استخدام الريليه الميكانيكى وهذا لأنه يتكون من ملف و نقط تلامس تعمل ميكانكياً فلذلك قد يتطلب وقت بين الإشارات حتى تقوم نقط التلامس بالفتح و بالغلق ولذلك فإذا كان الـــ PLC يصدر أشارات سريعة فيفضل إذاً استخدام الريليه الإلكترونى solid state relay عوض عن الريليه الميكانيكى حيث أنه يتميز بنقاط مساعدة تقوم بتغير الحالة بسرعة فائقة .

كيفية توصيل وحدة الـــ PLC بالكامل:

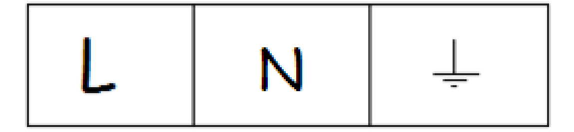
ينقسم التوصيل إلى ثلاث أجزاء:

(تغذية وحدة الـــ CPU – تغذية مجموعة المدخلات – تغذية مجموعة المخرجات)

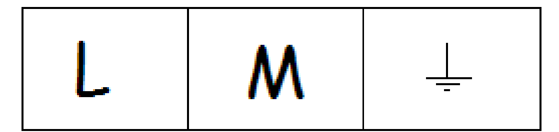
١- تغذية الـــ CPU.

تغذية الـــ CPU تطلب فقط توصيل الكهرباء حسب نوع الـــ PLC (مثلاً التيار المتردد أو التيار المستمر).

تيار متردد



تيار مستمر

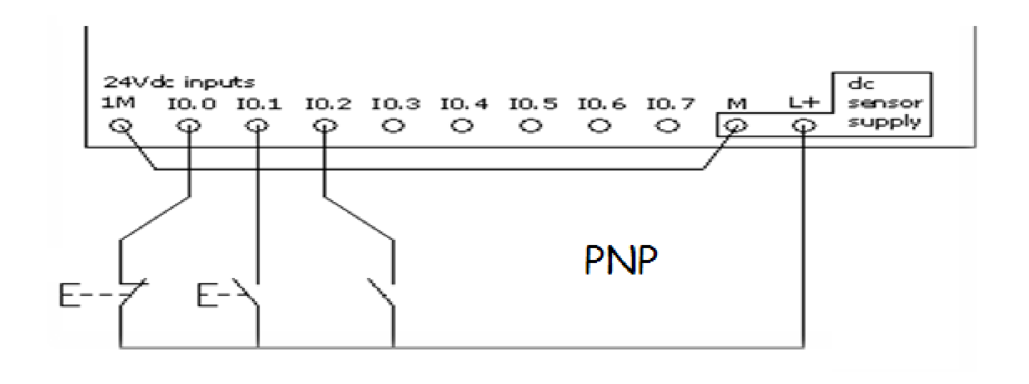


٢– تغذية مجموعة المدخلات.

كما سبق وشرحنا, تغذية مجموعة المدخلات تتطلب أولاً توصيل طرف من الكهرباء على المفاتيح وثانياً توصيل الطرف الأخر على وحدة المدخلات, قد يختلف التوصيل قليلاً حسب نوع الـــ PLC (PNP أو NPN).

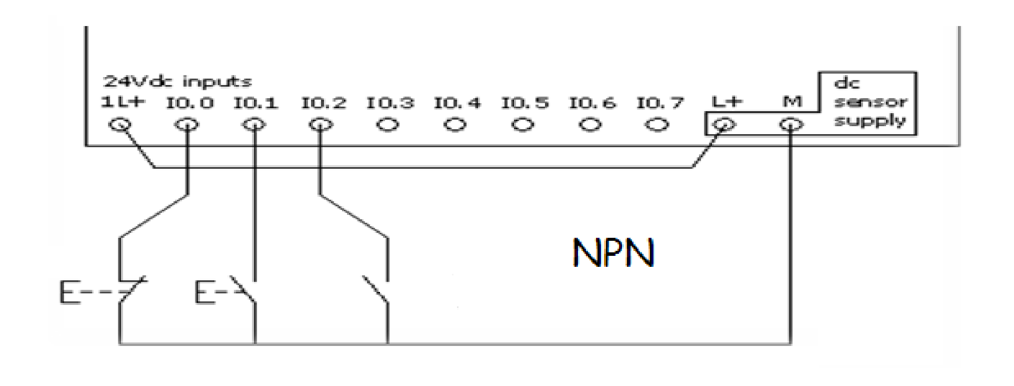
النوع الأول. PNP:

يتم توصيل الطرف الموجب على المفاتيح بينما يوصل الطرف السالب على وحدة الدخل نفسها.



النوع الثانى. NPN

يتم توصيل الطرف السالب على المفاتيح بينما يوصل الطرف الموجب على وحدة الدخل نفسها.



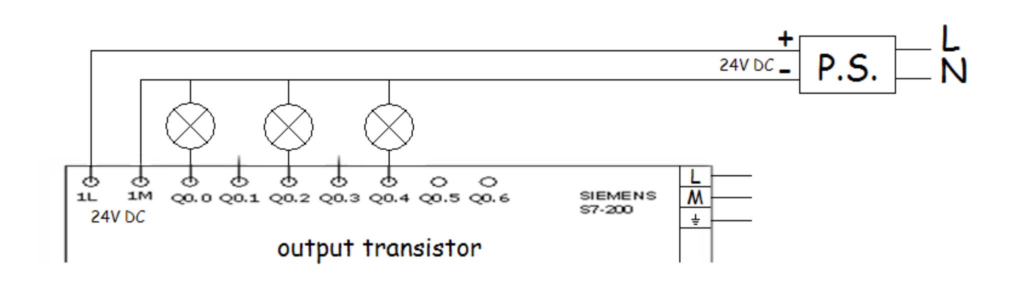
٣- تغذية مجموعة المخارج.

تغذية مجموعة المخرجات تتطلب فقط توصيل الكهرباء حسب نوع الـــ PLC (مثلاً transistor أو relay).

النوع الأول. Transistor:

فى حالة الـــ output transistor يتم استخدام التيار المستمر فقط ولذلك يعوض عن كلمة transistor بكلمة DC

ودائماً ما يكون الجهد الخاص بالخرج هو 24V DC كما هو مبين فر الرسم.



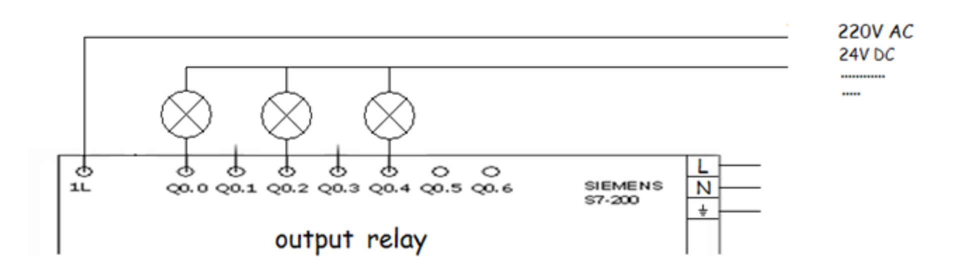
النوع الثانى. Relay:

فى حالة الـــ output relay يتم استخدام التيار المستمر أو التيار المتردد, يعوض عن كلمة relay بكلمة RLY

الجهد الخاص بالخرج هو متغير حيث يمكن توصيل أى قيمة ضمن الحد المسموح به من جهاز الـــ PLC.

مثلاً: (24V DC – 220V AC - ……..V DC - …….V AC)

كما هو مبين بالرسم.



كيف أعرف نوع الــ PLC.

يوجد نوعين من حيث طريقة التوصيل:

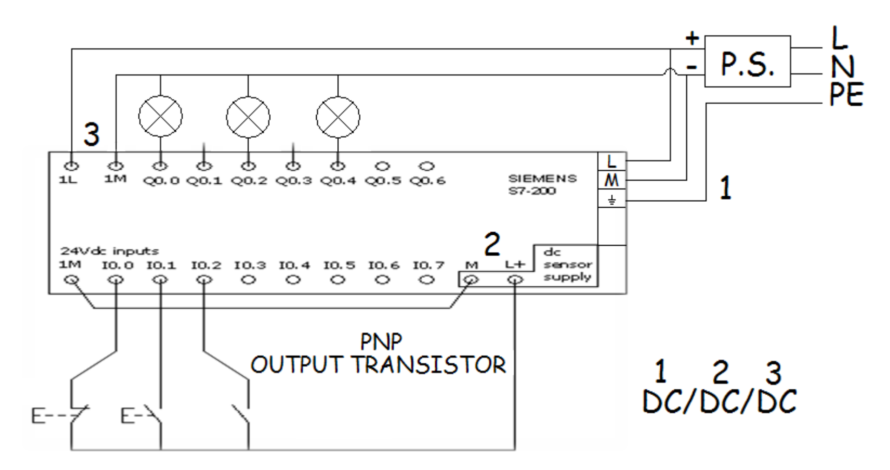
AC/DC/RLY - **DC/DC/DC**

النوع الأول. DC/DC/DC:

- كلمة **DC** الأولى تخص تغذية الــ CPU.
- كلمة DC الثانية تخص تغذية المداخل.
- كلمة DC الثالثة تخص تغذية المخارج.

المقصود بــ **DC** الأولى هى كلمة **DC** التى فى أقصى اليسار, أنظر الشكل التالى.

رسم توضيحى للنوع الأول **DC/DC/DC**.



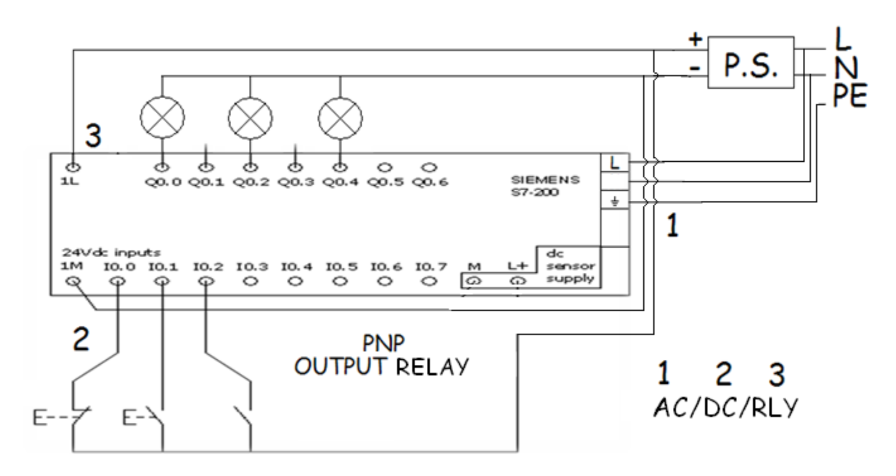
النوع الثانى. AC/DC/RLY:

كلمة **AC** تخص تغذية الـــ CPU.

كلمة **DC** تخص تغذية المداخل.

كلمة **RLY** تخص تغذية المخارج.

رسم توضيحى للنوع الثانى.



ملاحظة:

- جميع الموديلات يتوفر منها النوعين: DC/DC/DC و AC/DC/RLY.
- الـ output transistor هو DC/DC/DC.
- الـ output relay هو AC/DC/RLY.

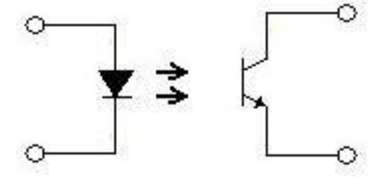
الأمثال للنوعين:

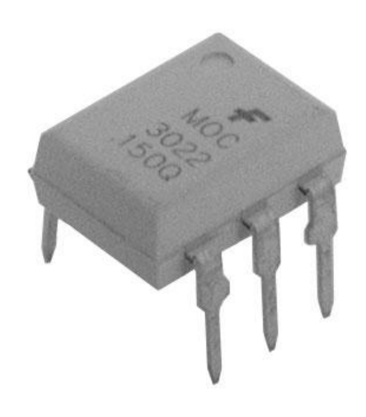
Model Description	Power Supply	Input Types	Output Types
221 DC/DC/DC	20.4-28.8 VDC	6 DC Inputs	4 DC Outputs
221 AC/DC/Relay	85-264 VAC 47-63 Hz	6 DC Inputs	4 Relay Outputs
222 DC/DC/DC	20.4-28.8 VDC	8 DC Inputs	6 DC Outputs
222 AC/DC/Relay	85-264 VAC 47-63 Hz	8 DC Inputs	6 Relay Outputs
224 DC/DC/DC	20.4-28.8 VDC	14 DC Inputs	10 DC Outputs
224 AC/DC/Relay	85-264 VAC 47-63 Hz	14 DC Inputs	10 Relay Outputs
226 DC/DC/DC	20.4-28.8 VDC	24 DC Inputs	16 DC Outputs
226 AC/DC/Relay	85-264 VAC 47-63 Hz	24 DC Inputs	15 Relay Outputs

حمايات داخلية للـــ PLC.

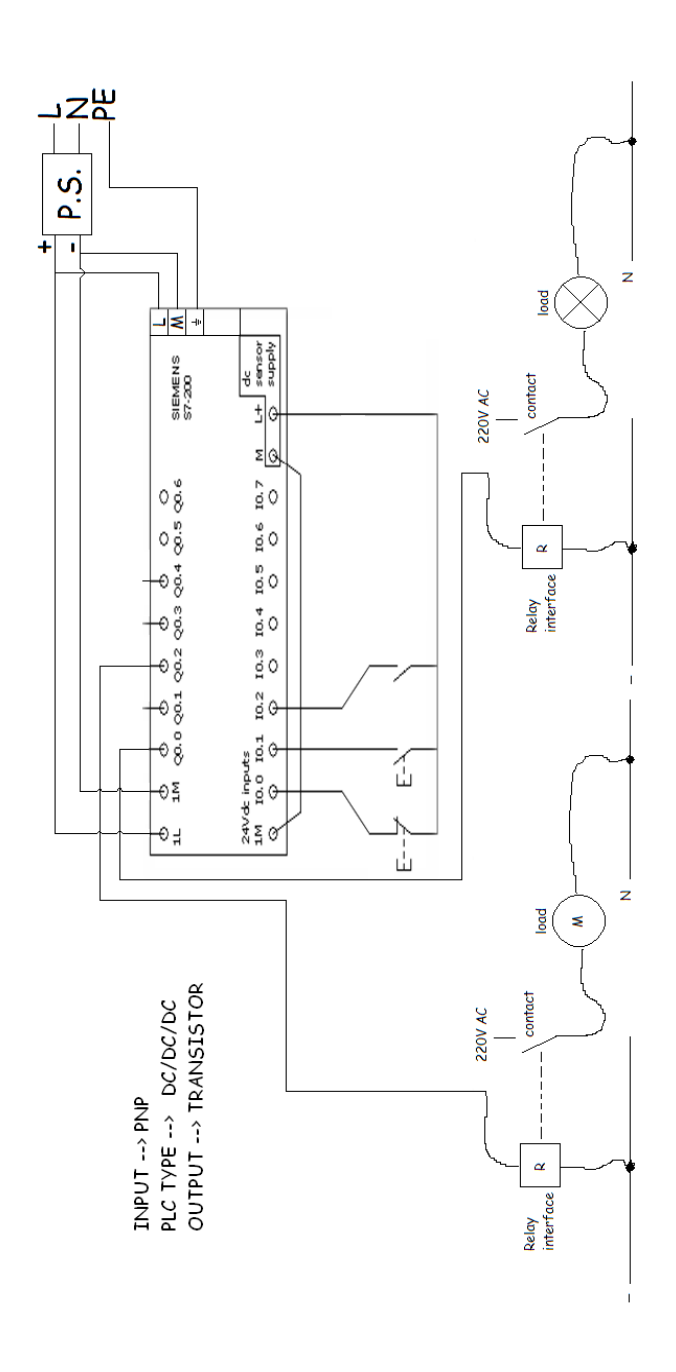
عازل كهروضوئى

يوجد داخل جهاز الـــ PLC عازل كهروضوئى optical coupler خاص بكل دخل (input) بحيث فى حالة توصيل أى جهد زيادة عن طريق الخطاء فأنه فى الغالب سوف يتضرر العازل الكهروضوئى وليس الـــ PLC وذالك لأن العازل الكهروضوئى يستعمل كعازل بين دائرتين كهربائيتين, حيث الإشارة الكهربائية القادمة من الدارة الأولى تتحول إلى إشارة ضوئية ومن ثم تقوم الدارة الثانية بتحويل هذا الإشارة الضوئية إلى كهربائية مرة أخرى, يعمل هذا العازل مع الإشارات الرقمية وكذلك مع التماثلية أو التناظرية .





الشكل العام للـــ PLC :



بالنظر لجهاز الــ PLC سوف يلاحظ:

١ – لمبات إشارة.

٢ – مفتاح التحكم بجهاز الــ PLC.

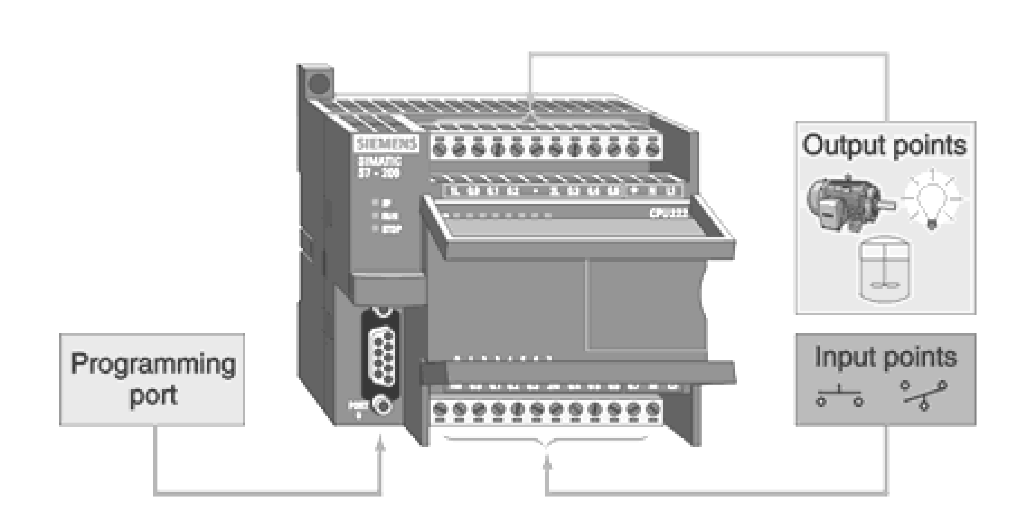
٣ – مفتاح الضبط التناظرى Analog adjustment.

٤ - مدخل كابل البرمجة.

٥ – مكان البطارية.

٦ – مكان الذاكرة.

٧ – مدخل وحدة دخل أو خرج إضافية Extension module.



١- لمبات الإشارة.....Indicators.

تقسم إلى ثلاث لمبات:

Run - Stop - S.F.

- لمبة **RUN**:

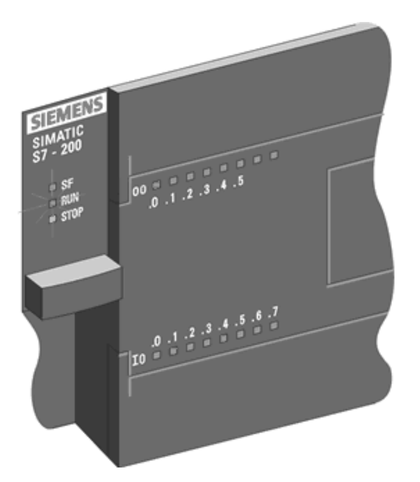
لمبة run تضاء لون أخضر حينما يعمل الــ PLC, أى عندما يكون الجهاز فى وضع run.

- لمبة STOP:

لمبة stop تضاء لون أخضر حينما يقف الــ PLC أى عندما يكون الجهاز فى وضع stop.

- لمبة SYSTEM FAULT:

لمبة system fault تضاء لون أحمر حينما يقف الــ PLC بسبب مشكلة ما.



ملاحظة:

فى حالة أن لمبة الــ .S.F مضاءة من المؤكد أن لمبة الــ stop ستكون مضاءة أيضاً نظراً لأن فى حالة وجود مشكلة فى الــ PLC ليس من الممكن أن يستمر فى العمل أى أنه ليس من المنطقى أن يبقى فى وضع run.

٢– مفتاح التحكم بجهاز الــ PLC......Mode Switch:

توجد ثلاث أوضاع لمفتاح التحكم بجهاز الــ PLC

Run - Stop - Terminal

- RUN:

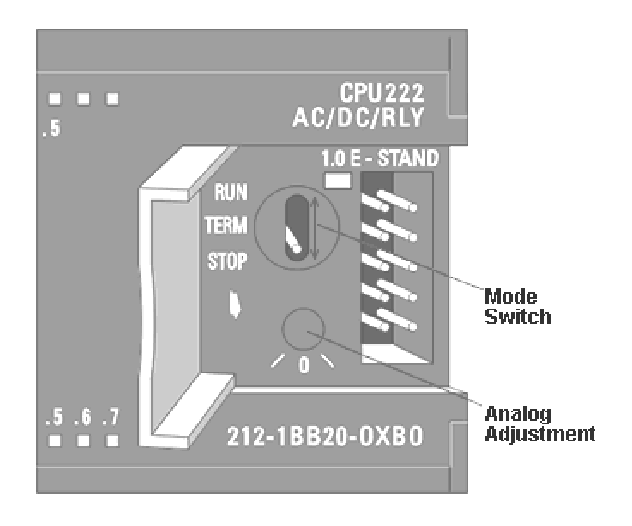
في حالة ضبط المفتاح على وضعية run يعمل جهاز الــ PLC أوتوماتيكياً وتضئ لمبة run بالون الأخضر.

- STOP:

في حالة ضبط المفتاح على وضعية stop يقف جهاز الــ PLC أوتوماتيكياً وتضئ لمبة stop بالون الأخضر.

- TERMINAL:

في حالة ضبط المفتاح على وضعية terminal يصبح من الممكن التحكم فى جهاز الــ PLC أوتوماتيكياً عن طريق الكمبيوتر وتضئ لمبة stop أو لمبة run حسب الأختيار.



٣– مفتاح ضبط الإشارة التناظرية.....Analog adjustment:

مفتاح ضبط الإشارة التناظرية يستخدم كمثال للدخل التناظرى حيث من الممكن تغير قيمة المفتاح بواسطة مفك لكى يستخدم فى البرمجة (يمكن تغير القيمة من صفر إلى ٢٥٥).

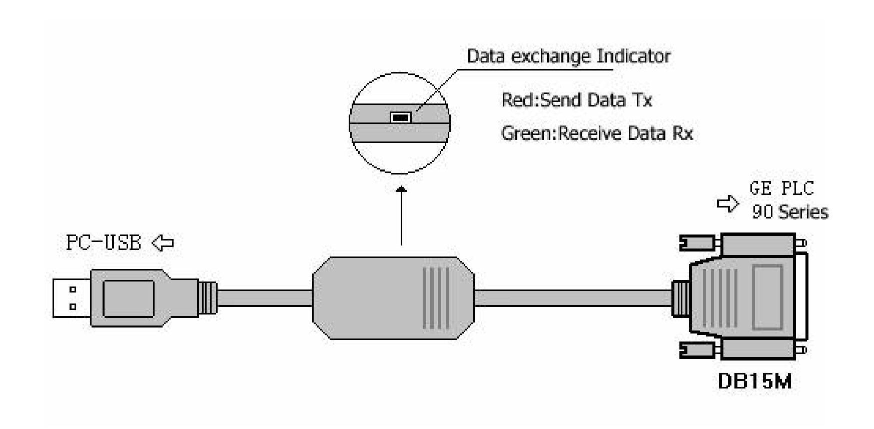
٤- مدخل كابل البرمجة.........Programming Cable:

حيث يتم توصيل الكابل بين جهاز الـــ PLC و جهاز البرمجة وهو الكمبيوتر فى هذه الحالة. و يسمى التوصيل بينهم بأسم (PPI (Point to Point Interface ويستخدم الكابل لنقل المعلومات من و إلى الـــ PLC.

يختلف نوع الكابل من بين بعض الأجهزة الموجودة بالسوق:

١ - كابل ذات سرعة 9.6Kbps
٢– كابل ذات سرعة 19.2Kbps
٣ -كابل ذات سرعة 187.5Kbps

-الكابل ذات سرعة 9.6Kbps يستطيع أن ينقل معلومات بحجم 9.6Kb فى ثانية واحدة فقط.
-الكابل ذات سرعة 19.2Kbps يستطيع أن ينقل معلومات بحجم 19.2Kb فى ثانية واحدة فقط.
وهكذا......



ملاحظة:

الكابل الخاص بالـ S7 200 توجد به ثلاث لمبات إشارة (Data exchange Indicator):

PPI - Tx - Rx

-اللمبة الأولى PPI.

تسمى PPI وهي أختصار لكلمة Point to Point Interface وهى تضاء بصفة مستمرة فى حالة أتصال جهاز الــ PLC بجهاز التحكم (computer) وتضاً بالون الأصفر.

-اللمبة الثانية Tx.

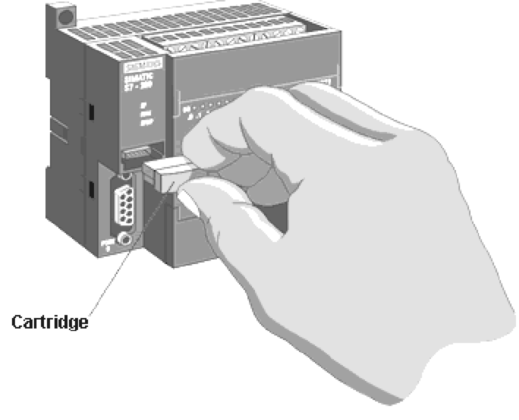
تسمى Tx وهي أختصار لكلمة Transfer Data وهى تضاء بصفة متقطعة أثناء أرسال معلومات إلى جهاز الــ PLC وتضاً بالون الأصفر.

-اللمبة الثالثة Rx.

تسمى Rx وهي أختصار لكلمة Receive Data وهى تضاء بصفة متقطعة أثناء إستقبال معلومات من جهاز الــ PLC وتضاً بالون الأصفر.

٥– البطارية.......Battery:

يتم توصيل البطارية في جهاز الــ PLC لكي تعمل كمصدر مستمر للتيار لهدف الحفاظ على البرنامج داخل الذاكرة حتى فى حالة انقطاع المصدر الرئيسى للتغذية (الكهرباء).



ملاحظة:

من أكثر أنواع البطاريات المستخدمة هو Lithium:

- جهد البطارية هو 3.6V
- يفضل تغيير البطارية كل سنتين.

المميزات:

- التيار (ampere curve) الخاص بالبطارية ثابت.

العيوب:

- التيار (ampere curve) الخاص بالبطارية يترل إلى صفر عند نهاية العمر الإفتراضى للبطارية دون سابق أنظار.



٦- <u>الذاكرة.......Memory</u>:

تنقسم الذاكرة فى جهاز الــ PLC إلى نوعين:

-ذاكرة موجودة داخل الــ PLC و دائماً تكون من النوع المتطاير volatile memory.

-ذاكرة تضاف إلى الــ PLC و دائماً تكون من النوع الغير المتطاير non volatile memory.

-النوع الأول.

<u>ذاكرة متطايرة</u>.......RAM (Random Access Memory)

مميزات الذاكرة المتطايرة

يمكن رسم, تعديل و مسح البرنامج أكثر من مرة.

عيوب الذاكرة المتطايرة

فى حالة أنقطاع التيار يمسح البرنامج بالكامل (إذا كان بدون بطارية).

ملاحظة:

- لحل مشكلة إنقطاع التيار يتم توصيل بطارية.

-النوع الثانى.

<u>ذاكرة غير متطايرة</u>........ROM (Read Only Memory):

مميزات الذاكرة غير المتطايرة

فى حالة إنقطاع التيار لا يمح البرنامج (حتى إذا كان بدون بطارية).

عيوب الذاكرة غير المتطايرة

لا يمكن تعديل أو مسح البرنامج مرة أخرى بعد تنفيذه.

ملاحظة:

- توجد أنواع أخرى من مشتقات الـــ ROM وهى:

<u>EPROM</u> - **<u>EEPROM</u>** - **<u>FLASH</u>** **<u>MEMORY</u>**

-النوع الأول.

ذاكرة غير متطايرة

EPROM (Electrical Programmable Read Only Memory)

- كيفية كتابة البرنامج:

يتم نقل البرنامج من الكمبيوتر إلى الذاكرة الـــ EPROM بواسطة كرت برمجة حيث يضع بداخلة الذاكرة ويوصل كرت البرمجة بالكمبيوتر, و بالضغط على "تحميل" يتم تحميل البرنامج على الذاكرة.

- كيفية مسح البرنامج:

يتم مسح البرنامج من على هذا النوع من الذاكرة عن طريق الأشعة الفوق البنفسجية بحيث يتم وضع الذاكرة داخل جهاز الأشعة الفوق البنفسجية لزمن محدد ويتم تنفيذ هذه العملية من قبل شخص ذات خبرة لتجنب تلف الذاكرة.

-النوع الثانى.

ذاكرة غير متطايرة

EEPROM (Erasable Electrical Programmable Read Only Memory)

- كيفية كتابة البرنامج:

يتم نقل البرنامج من الكمبيوتر إلى الذاكرة الـــ EPROM بواسطة كرت برمجة حيث يضع بداخلة الذاكرة ويوصل كرت البرمجة بالكمبيوتر, بالضغط على "تحميل" يتم تحميل البرنامج على الذاكرة.

- كيفية مسح البرنامج:

يتم نقل البرنامج من الكمبيوتر إلى الذاكرة الــ EPROM بواسطة كرت برمجة حيث يضع بداخلة الذاكرة ويوصل كرت البرمجة بالكمبيوتر, و بالضغط على erase memory cartridge يتم مسح جميع محتويات الذاكرة.

-النوع الثالث.

<u>ذاكرة غير متطايرة</u>

FLASH MEMORY

- كيفية كتابة البرنامج:

يمكن بكل سهولة كتابة البرنامج على هذا النوع من الذاكرة دون أى مشكلة و دون استخدام كرت البرمجة أى التحكم بالذاكرة بدون فصلها عن جهاز الــ PLC.

- كيفية مسح البرنامج:

يمكن بكل سهولة مسح البرنامج من على هذا النوع من الذاكرة دون أى مشكلة و دون استخدام كرت البرمجة أى التحكم بالذاكرة بدون فصلها عن جهاز الــ PLC.

ملاحظة:

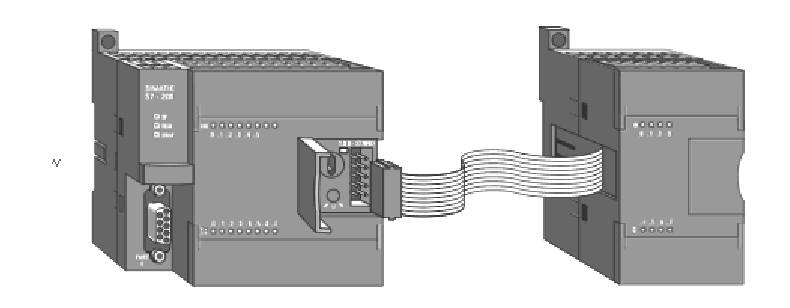
- قبل كتابة البرنامج على أى ذاكرة من نوع

EPROM أو EEPROM أو FLASH MEMORY

يجب تجربة البرنامج على الذاكرة الــ RAM عدة مرات للتأكد من عدم وجود أى مشاكل بالبرنامج لأن كثرة المسح أو التعديل على الذاكرة الغير متطيرة يؤثر على العمر الإفتراضى للذاكرة.

٧ - وحدات دخل أو خرج إضافية.........**Extension Module**:

نظراً لأن فى بعض الحالات قد يحتاج المبرمج إلى مجموعة مداخل أو مخارج إضافية لاستخدامها فى البرمجة فأنة من الممكن شراء وحدات تحتوى على عدد محدد من المداخل فقط أو وحدات تحتوى على عدد محدد من المخارج فقط أو وحدات تحتوى على عدد محدد من مداخل و المخارج معاً.



مثال لوحدات الدخل و الخرج الإضافية:

-وحدات دخل و خرج رقمية إضافية:

١- EM221: وحدة إضافية تحتوى على ثمانية مداخل.

٢- EM222: وحدة إضافية تحتوى على ثمانية مخارج.

٣ - EM223: وحدة إضافية تحتوى على أربعة مداخل و أربعة مخارج.

٤ - EM223: وحدة إضافية تحتوى على ثمانية مداخل و ثمانية مخارج.

٥- EM223: وحدة إضافية تحتوى على ستة عشر دخل و ستة عشر خرج.

-وحدات دخل و خرج تناظرية إضافية:

١- EM231: وحدة إضافية تحتوى على أربعة مداخل.

٢- EM232: وحدة إضافية تحتوى على مخرجان.

٣ - EM235: وحدة إضافية تحتوى على أربعة مداخل و خرج واحد.

ملاحظة:

يختلف توصيل الوحدات الإضافية بين بعض أجهزة الـــ PLCS, فمثلاً:

- فى حالة توصيل وحدة أضافية لجهاز PLC CPU 224 يتم توصيل الوحدة الإضافية إلى جهاز الـــ PLC عن طريق كابل.
- فى حالة توصيل وحدة إضافية لجهاز PLC CPU 214 يتم توصيل الوحدة الإضافية إلى جهاز الـــ PLC عن طريق تركيب الاثنان معاً أى بالتوصيل المباشر.

يفضل توصيل الوحدات الإضافية إلى وحدة الـــ PLC بواسطة الكابل وهذا لأنه يمكن وضع الاثنين تحت بعضهم أو على مسافة بعيدة.

أجهزة للتحكم فى وحدة الـــ PLC:

من الممكن التحكم فى جهاز الـــ PLC عن طريق:

Computer - HMI

أى أنه يمكن التحكم بوحدة التحكم المنطقى باستخدام شاشة التحكم أو باستخدام الكمبيوتر ويمكن أيضاً استخدام الاثنين معاً

١– الكمبيوتر........Computer:

حيث يتم بسهولة عمل أى برنامج بأى لغة برمجة, تحميل البرنامج إلى جهاز الــ PLC, أخذ البرنامج من جهاز الــ PLC, التعديل فى البرنامج بالأضافة أو بالمسح, معرفة حالة البرنامج و كيف يعمل و تغير بعض البيانات دون أقاف البرنامج....الخ



٢– شاشة التحكم........(Human Machine Interface):

حيث يتم بسهولة التعديل فى البرنامج فى حدود معينة, معرفة حالة البرنامج و كيف يعمل, تغير بعض البيانات دون أقاف البرنامج, رؤية الألرمات على الشاشة و معرفة نوع العطل, أطفاء الألرمات....الخ



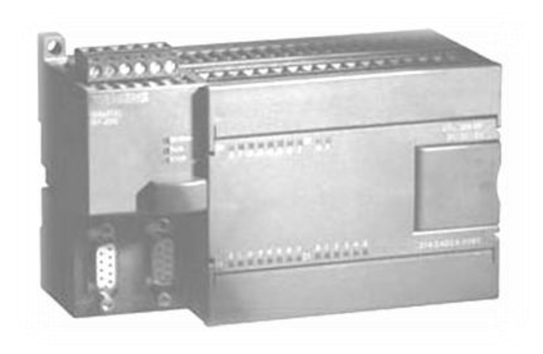
ملاحظة:

١- يستخدم جهاز الكمبيوتر فى البداية فقط لعمل البرنامج بالكامل ويفضل بعد ذالك أستبدال الكمبيوتر بشاشة التحكم نظراً لصغر حجمها.

٢ - بعض أجهزة الــ PLC تحتوى على مخرج واحد للبرمجة مثل جهاز (PLC (CPU 224 والبعض الأخر يحتوى على مخرجين للبرمجة مثل جهاز (PLC (CPU 224 XP و لذالك فى الــ CPU 224 يتم توصيل أما جهاز كمبيوتر أما شاشة تحكم بينما فى الــ CPU 224 XP من الممكن توصيل جهاز كمبيوتر و شاشة تحكم معاً.

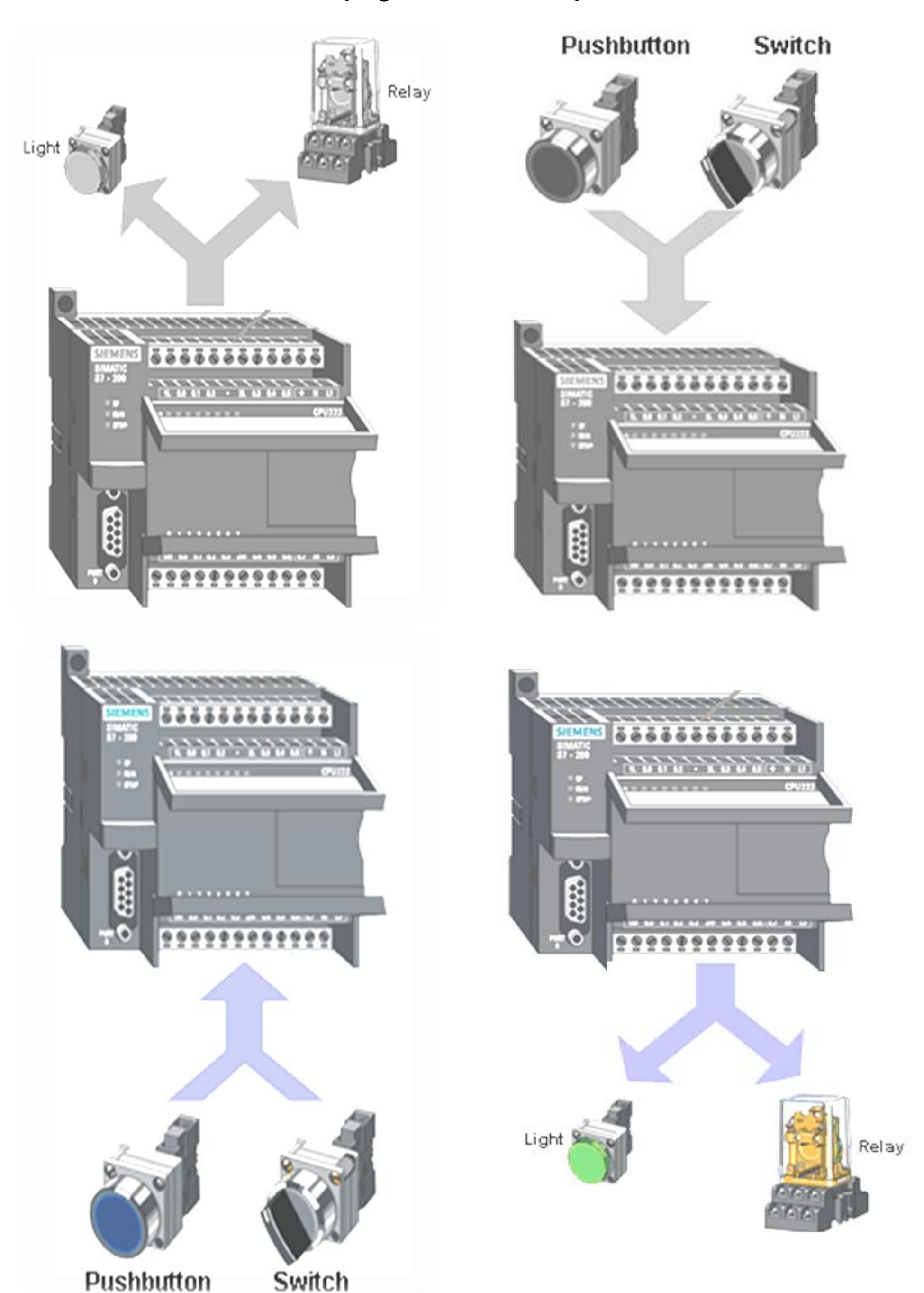


CPU 224



CPU 224 XP

ملاحظة هامة: يختلف مكان المدخلات و المخرجات بين بعض الموديلات.



الباب الثانى

الذاكرة

- تمهيـــــــــــد عن الذاكرة داخل جهاز الــ PLC.
- أحجام الذاكــــــرة المستخدمة فى جهاز الــ PLC.
- كيفية كتابة بيانات داخل الذاكـــــــــــــــــــرة.
- كيفية قراءة بيانات من الذاكـــــــــــــــــــرة.
- نظـــــــــــــــــــــــــــام الأعداد الثنائية.
- نظـــــــــــــــــــــــــــام الأعداد العشرية.
- نظـــــــــــــــــــــــــام الأعداد السداسية عشر.
- نظـــــــــــــــــــام الأعداد العشرية المكودة ثنائياً.
- نظـــــــــــــــــــــــــــام العلامة العشرية.
- العمليات الحسابية للأعداد الثنائيـــــــــــــــــــة.
- كـــيـــفـــيـــة التحـــويـــل من نظـــــام إلى أخر.

أحجم الذاكرة المستخدمة فى جهاز الــ PLC :

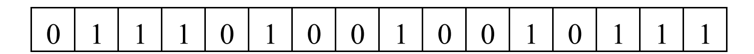
١- BIT: هى أصغر وحدة للذاكرة داخل جهاز الــ PLC وهى قد تحتوى على صفر أو واحد.

1

٢- BYTE: هى ذاكرة داخل جهاز الــ PLC وهى قد تحتوى على صفر و واحد وهى تتكون من 8 bits.

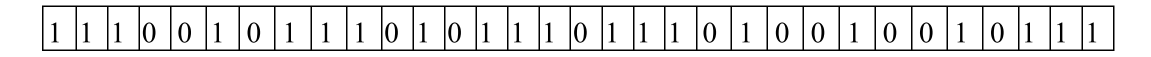
1	0	1	0	0	0	1	1

٣- WORD: هى ذاكرة داخل جهاز الــ PLC وهى قد تحتوى على صفر و واحد وهى تتكون من 2 bytes أو 16 bits.



0	1	1	1	0	1	0	0	1	0	0	1	0	1	1	1

٤- D.WORD: هى أكبر وحدة للذاكرة داخل جهاز الــ PLC وهى قد تحتوى على صفر و واحد وهو تتكون من 2 words أو 4 bytes أو 32 bits.



1	1	1	0	0	1	0	1	1	1	0	1	0	1	1	1	0	1	1	1	0	1	0	0	1	0	0	1	0	1	1	1

هام:

لكل وحدة من الذاكرة أسم و طريقة للكتابة:

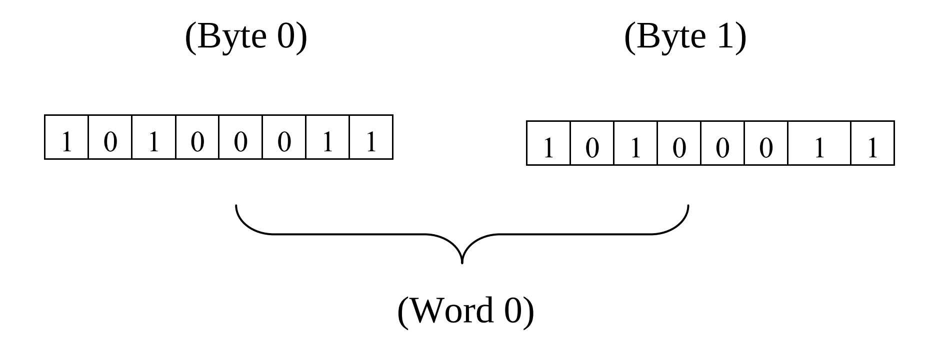
- BIT 0, BIT 1, BIT 2, BIT 3, BIT 4, ...
- BYTE 0, BYTE 1, BYTE 2, BYTE 3, ...
- WORD 0, WORD 2, WORD 4, WORD 6, ...
- D.WORD 0, D.WORD 4, D.WORD 8, D.WORD 12

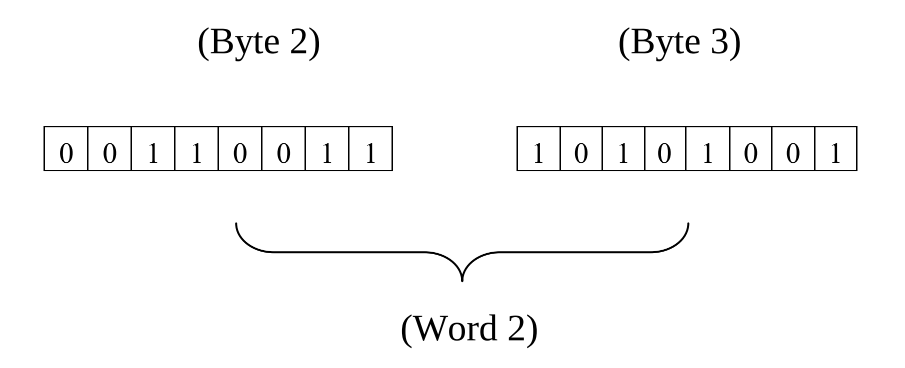
ملاحظة:

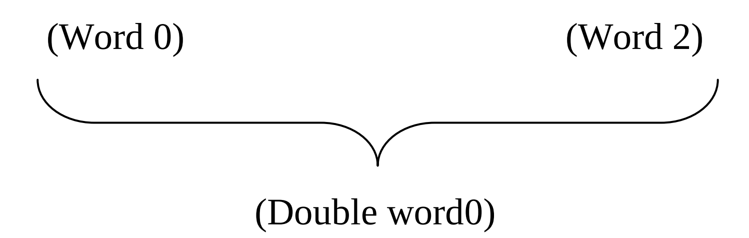
من المهم جداً مراعاة أن فى حالة كتابة معلومات بواسطة الـــ bit يجب تحديد أسم الـــ bit.

فى حالة كتابة معلومات بواسطة الـــ byte أو الـــ word أو الـــ Dword يجب بدء الكتابة من اليمين إلى اليسار وهذا لأن جميع أحجام الذاكرة هى مكونة من مجموعة من الـــ bits وتستخدم نفس الطريقة أيضاً فى حالة قراءة البيانات من على الذاكرة.

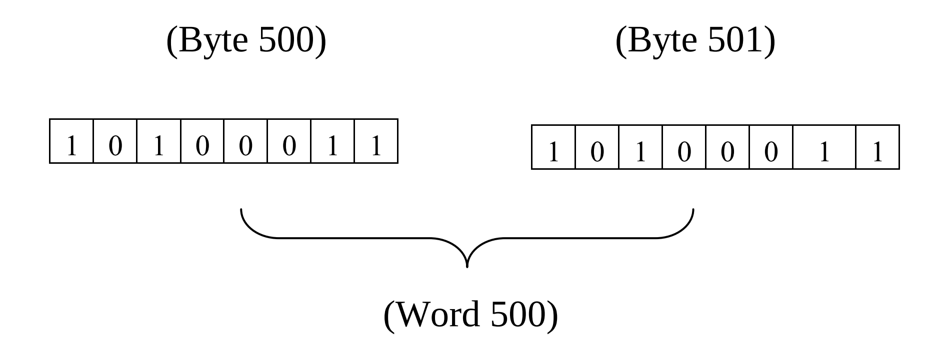
شرح مفصل:

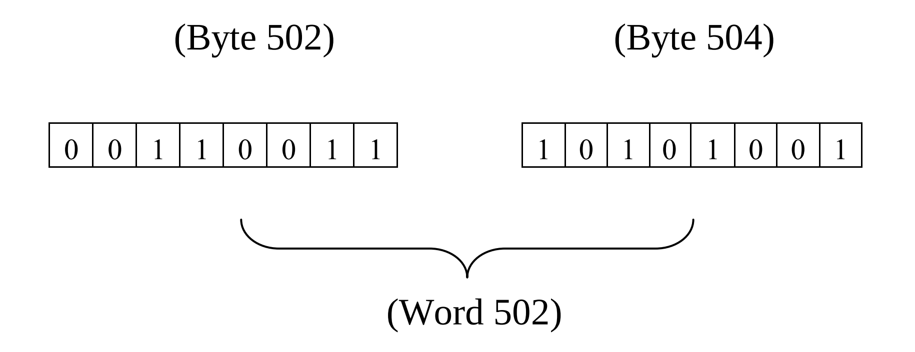


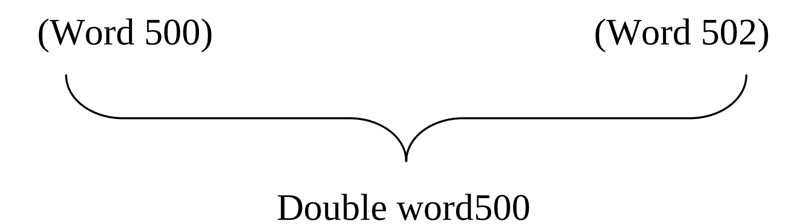



(Word 0)
(Word 2)
(Double word0)

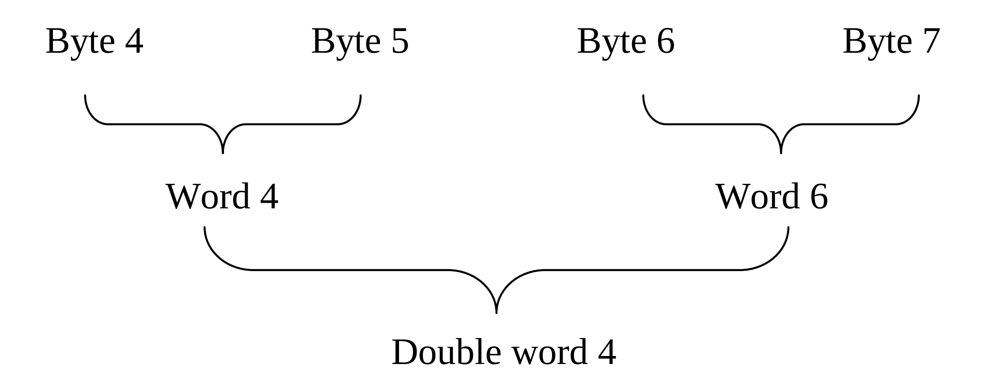
مثال أخر:


(Byte 500)
(Byte 501)
1 0 1 0 0 0 1 1
1 0 1 0 0 0 1 1
(Word 500)

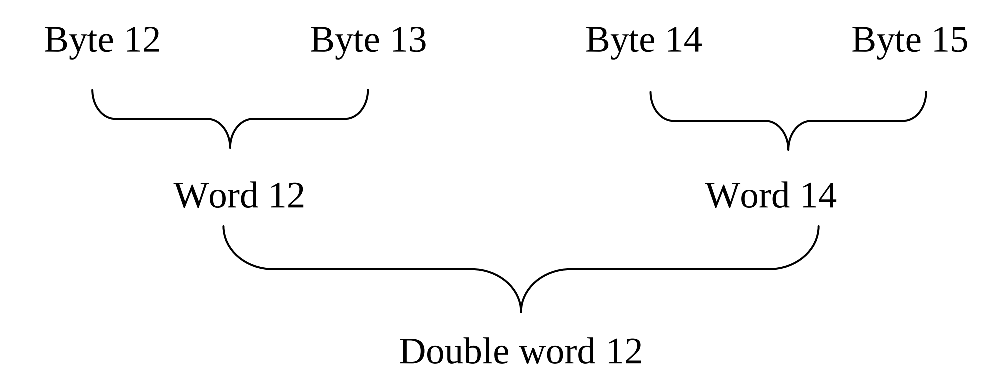

(Byte 502)
(Byte 504)
0 0 1 1 0 0 1 1
1 0 1 0 1 0 0 1
(Word 502)



مثال أخر:



مثال أخر:



تكوين الــ byte من الداخل:

bit0.7 bit0.6 bit0.5 bit0.4 bit0.3 bit0.2 bit0.1 bit0.0

Byte 0

مثال أخر

bit11.7 bit11.6 bit11.5 bit11.4 bit11.3 bit11.2 bit11.1 bit11.0

Byte 11

كيفية قراءة ما بداخل الذاكرة (حسب المثال المذكور فى صفحة 61):

- فى حالة قراءة أى bit تتم القراءة من اليمين إلى اليسار ولكن لذكر أسم الــ bit يجب تحديد لأى byte ينتمى هذا الــ bit فمثلاً: b2.0 هو الــ bit الأول فى الــ byte الثالث أو الــ b5.6 هو الــ bit السابع فى الــ byte السادس.

- فى حالة قراءة أى byte من الداخل تتم القراءة من اليمين إلى اليسار لأن الــ byte يتكون من مجموعة bits وبعد قراءة ذالك الــ byte تتم قراءة الــ byte الذى على يمينه بنفس الطريقة أى أن فى حالة قراءة البيانات من على مجموعة bytes فمثلاً يقرأ الــ byte 0 ثم يقرأ الــ byte الذى على يمينة الذى هو byte 1 وهكذا.

- فى حالة قراءة أى word من الداخل تتم القراءة من اليمين إلى اليسار لأن الـــ word يتكون من مجموعة bits وبعد قراءة ذالك الـــ word تتم قراءة الـــ word الذى على يمينه بنفس الطريقة أى أن فى حالة قراءة البيانات من على مجموعة words فمثلاً يقرأ الـــ word 0 ثم يقرأ الـــ word الذى على يمينه الذى هو word 2 وهكذا وبالنسبة إلى الـــ word 2 يقرأ الـــ bits من اليمين إلى اليسار أى يقرأ الـــ byte 3 ثم الـــ byte2.

- فى حالة قراءة أى D.word من الداخل تتم القراءة من اليمين إلى اليسار لأن الـــ D.word يتكون من مجموعة bits وبعد قراءة ذالك الـــ D.word تتم قراءة الـــ D.word الذى على يمينه بنفس الطريقة أى أن فى حالة قراءة البيانات من على مجموعة D.words فمثلاً يقرأ D.word0 ثم يقرأ الـــ D.word الذى على يمينه الذى هو D.word4 وهكذا.

بالنسبة إلى الـــ D.word0 يقرأ الـــ bits من اليمين إلى اليسار أى يقرأ الـــ byte3 ثم الـــ byte2 ثم الـــ byte1 ثم الـــ byte0 أى أنه يقرأ الـــ word2 ثم الـــ word0.

النظم الرقمية المستخدمة فى جهاز الـــ PLC :

١- نظام الأعداد الثنائية (binary).

٢- نظام الأعداد العشرية (decimal).

٣ - نظام الأعداد السداسية عشر (hexadecimal).

٤ - نظام الأعداد العشرية المكودة ثنائياً (BCD).

٥ - نظام العلامة العشرية (real).

٦ - نظام العلامة العشرية (floating point).

٧ - نظام أسكى (ASCII).

١– النظام الثنائى (binary).

يتكون من:

(صفر) و (١).

مثال:

.$(10010111)_2$

لذلك لكتابة أرقام من نوع النظام الثنائى binary يمكن استخدام ذاكرة بحجم:

- BIT.
- BYTE.
- WORD.
- DOUBLE WORD.

٢– النظام العشرى (decimal).

يتكون من:

(صفر), (١), (٢), (٣), (٤), (٥), (٦), (٧), (٨) و (٩).

مثال:

.$(55)_{10}$

لذلك لكتابة أرقام من نوع النظام العشرى decimal **يمكن استخدام ذاكرة بحجم:**

- BYTE.
- WORD.
- DOUBLE WORD.

٣ - النظام السداسى عشر (hexadecimal).

يتكون من:

(صفر), (١), (٢), (٣), (٤), (٥), (٦), (٧), (٨) , (٩), (A), (B), (C), (D), (E) و (F).

مثال:

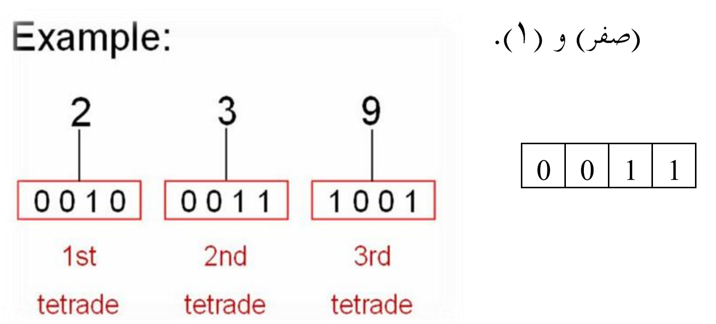
.$(6A2F)_{16}$

لذلك لكتابة أرقام من نوع النظام السداسي عشر hexadecimal **يمكن أستخدام ذاكرة بحجم:**

- BYTE.
- WORD.
- DOUBLE WORD.

٤ - نظام الأعداد العشرية المكودة ثنائياً (**Binary Code Decimal**).


يتكون من:



(صفر) و (١).

مثال:

0	0	1	1

لذلك لكتابة أرقام من نوع نظام الــ BCD **يمكن أستخدام ذاكرة بحجم:**

- BYTE.
- WORD.
- DOUBLE WORD.

٥ - نظام العلامة العشرية (**real**).

يتكون من:

(صفر), (١), (٢), (٣), (٤), (٥), (٦), (٧), (٨) و (٩).

مثال:

52.14

لكتابة أرقام من نوع نظام العلامة REAL **يمكن أستخدام فقط ذاكرة بحجم:**

- DOUBLE WORD.

٦ - نظام العلامة العشرية (floating point).

يتكون من:

(صفر), (١), (٢), (٣), (٤), (٥), (٦), (٧), (٨) و (٩).

مثال:

+5.2120000 + E 1

لكتابة أرقام من نوع نظام العلامة FLOATING **يمكن أستخدام فقط ذاكرة بحجم:**

- DOUBLE WORD.

٧- نظام أسكى (American Standard Code for Information Interchange).

هو جدول يحتوى على جميع الأرقام و الحروف و الرموز التى يمكن كتابتها بواسطة لوحة المفاتيح كما هو موضح بالصورة, لأن الــ PLC لا يفهم سوى الصفر و الواحد.

الجدول:

ASCII Code: Character to Binary

0	0011 0000	O	0100 1111	m	0110 1101
1	0011 0001	P	0101 0000	n	0110 1110
2	0011 0010	Q	0101 0001	o	0110 1111
3	0011 0011	R	0101 0010	p	0111 0000
4	0011 0100	S	0101 0011	q	0111 0001
5	0011 0101	T	0101 0100	r	0111 0010
6	0011 0110	U	0101 0101	s	0111 0011
7	0011 0111	V	0101 0110	t	0111 0100
8	0011 1000	W	0101 0111	u	0111 0101
9	0011 1001	X	0101 1000	v	0111 0110
A	0100 0001	Y	0101 1001	w	0111 0111
B	0100 0010	Z	0101 1010	x	0111 1000
C	0100 0011	a	0110 0001	y	0111 1001
D	0100 0100	b	0110 0010	z	0111 1010
E	0100 0101	c	0110 0011	.	0010 1110
F	0100 0110	d	0110 0100	,	0010 0111
G	0100 0111	e	0110 0101	:	0011 1010
H	0100 1000	f	0110 0110	;	0011 1011
I	0100 1001	g	0110 0111	?	0011 1111
J	0100 1010	h	0110 1000	!	0010 0001
K	0100 1011	I	0110 1001	'	0010 1100
L	0100 1100	j	0110 1010	"	0010 0010
M	0100 1101	k	0110 1011	(	0010 1000
N	0100 1110	l	0110 1100	)	0010 1001
				space	0010 0000

النظم الرقمية هامه جداً لأنه من الممكن قراءة أو كتابة البيانات بنظم رقمية مختلفة, فمثلاً:

- فى حالة التعامل مع دخل أو خرج من الأسهل أن تكون الصيغة format من نوع binary بحيث يكون من السهل تمييز حالة المفتاح أو الخرج, فمثلاً بالنسبة لحالة المفتاح أذا كان يساوى واحد فهذا يعنى أن المفتاح مغلق أما أذا كان يساوى صفر فهذا يعنى أن المفتاح مفتوح وهكذا أيضاً فى حالة الخرج و فى حالة الريليه الداخلى.

- فى حالة التعامل مع مؤقتات زمنية أو عدادات من الأسهل أن تكون الصيغة format من نوع decimal بحيث يكون من السهل قراءة أو كتابة أى زمن للمؤقت أو أى عدد للعداد.

التحويل من نظم رقمية إلى نظم رقمية أخرى مستخدمة فى جهاز الــ PLC :

- **من الــ decimal إلى الــ binary:**

يتم التحويل من الــ decimal إلى الــ binary بواسطة القسمة على أثنين, فمثلاً: رقم 100:

100 / 2 = 50	الباقى ⇨	0
50 / 2 = 25	الباقى ⇨	0
25 / 2 = 12	الباقى ⇨	1
12 / 2 = 6	الباقى ⇨	0
6 / 2 = 3	الباقى ⇨	0
3 / 2 = 1	الباقى ⇨	1
1	الباقى ⇨	1

$$(100)_{10} = (1100100)_2$$

يكتب المثال السابق بهذه الطريقة للتبسيط:

100	2	0	
50	2	0	
25	2	1	
12	2	0	**الباقي**
6	2	0	
3	2	1	
1	2	1	

تتم القسمة على أثنان لأن الأساس الخاص بنظام الأعداد الثنائية هو أثنان.

مثال أخر:

رقم $(255)_{10}$

255 / 2 = 127	الباقي ⇨	1
127 / 2 = 63	الباقي ⇨	1
63 / 2 = 31	الباقي ⇨	1
31 / 2 = 15	الباقي ⇨	1
15 / 2 = 7	الباقي ⇨	1
7 / 2 = 3	الباقي ⇨	1
3 / 2 = 1	الباقي ⇨	1
1	⇨	1

⇨ $(255)_{10} = (11111111)_2$

يكتب المثال السابق بهذه الطريقة للتبسيط:

255	2	1	
127	2	1	
63	2	1	
31	2	1	**الناتج**
15	2	1	
7	2	1	
3	2	1	
1	2	1	

بعض الأمثلة على التحويل من الـــ decimal **إلى الـــ** binary.

Binary numbers					
DECIAML		BINARY	DECIMAL		BINNARY
١	→	000000001	٥١	→	00110011
٢	→	00000010	٥٢	→	00110100
٣	→	00000011	٥٣	→	00110101
٤	→	00000100	٥٤	→	00110110
٥	→	00000101	٥٥	→	00110111
٦	→	00000110	٥٦	→	00111000
٧	→	00000111	٥٧	→	00111001
٨	→	00001000	٥٨	→	00111010
٩	→	00001001	٥٩	→	00111011
١٠	→	00001010	٦٠	→	00111100
١١	→	00001011	٦١	→	00111101
١٢	→	00001100	٦٢	→	00111110
١٣	→	00001101	٦٣	→	00111111
١٤	→	00001110	٦٤	→	01000000
١٥	→	00001111	٦٥	→	01000001
١٦	→	00010000	٦٦	→	01000010
١٧	→	00010001	٦٧	→	01000011
١٨	→	00010010	٦٨	→	01000100
إلخ		إلخ	إلخ		إلخ

- **من الــ binary إلى الــ decimal**:

لكل bit قيمة ثابتة, فمثلاً الــ bit0 قيمته تساوى 2^0 أى يساوى 1 أذا قمنا بنفس الشئ فى جميع الــ bits فالقيم سوف تكون:

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
المجموع الكلى لمحتوى الــ byte هو 255 فى هذه الحالة.							

يتم التحويل من الــ binary إلى الــ decimal بواسطة ضرب محتوى الــ bit فى القيمة الخاصة به, فمثلاً:

١ - رقم $(11011100)_2$
٢ - رقم $(11001110)_2$
٣ - رقم $(01001111)_2$

المثال الأول:

رقم $(11011100)_2$

1	**1**	**0**	**1**	**1**	**1**	**0**	**0**
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	0	16	8	4	0	0
المجموع الكلى لمحتوى الــ byte هو 220 فى هذه الحالة.							

المثال الثانى:

رقم $(11001110)_2$

0	**0**	**0**	**0**	**1**	**1**	**1**	**0**
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	0	0	8	4	2	0
المجموع الكلى لمحتوى الــ byte هو 206 فى هذه الحالة.							

المثال الثالث:

رقم $(01001111)_2$

0	**1**	**0**	**0**	**1**	**1**	**1**	**1**
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	64	0	0	8	4	2	1
المجموع الكلى لمحتوى الــ byte هو 79 فى هذه الحالة.							

- من الـــ **decimal** إلى الـــ **hexadecimal**:

يتم التحويل من الـــ decimal إلى الـــ hexadecimal بواسطة القسمة على 16, فمثلاً:

١- رقم: 100

٢- رقم: 50

المثال الأول:

رقم 100

100 / 16 = 6 الباقى ⇨ 4
6 = 6 الباقى ⇨ 6

⇨ $(100)_{10} = (64)_{16}$

يكتب المثال السابق بهذه الطريقة للتبسيط:

100	16	4	**الناتج**
6		6	

المثال الثانى:

رقم 50

50 / 16 = 3 الباقى ⇨ 2
3 = 3 الباقى ⇨ 3

⇨ $(50)_{10} = (32)_{16}$

يكتب المثال السابق بهذه الطريقة للتبسيط:

50	16	2	**الناتج**
3		3	

- **من الــ hexadecimal إلى الــ decimal**:

يتم التحويل من الــ hexadecimal إلى الــ decimal بواسطة الضرب فى 16, فمثلاً:

١- رقم 101A:

٢- رقم A0F:

المثال الأول:

رقم 101A

1	**0**	**1**	**A**
16^3	16^2	16^1	16^0
4096	0	16	10
4122			

المثال الثانى:

رقم 200F

2	**0**	**0**	**F**
16^3	16^2	16^1	16^0
8192	0	0	15
8207			

- **من الــ hexadecimal إلى الــ binary**:

يتم تحويل أى رقم hexadecimal إلى أربع أرقام binary, فمثلاً:

١- رقم 1A:

٢- رقم AF:

المثال الأول:

رقم 1A

1				**A**			
2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0
0	0	0	1	1	0	1	0
المجموع الكلى لمحتوى الــ byte هو: **00011010**							

المثال الثانى:

رقم AF

A				**F**			
2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0
1	0	1	0	1	1	1	1
المجموع الكلى لمحتوى الــ byte هو: **10101111**							

- **من الـــ binary إلى الـــ hexadecimal**:

يتم تحويل كل أربع أرقام رقم من الـــ binary إلى ما يعادلها في الـــ decimal ثم تكتب بصيغة الـــ hexadecimal.

١- رقم $(11001010)_2$:
٢- رقم $(00010111)_2$:

المثال الأول:

رقم $(11001010)_2$

1	1	0	0	1	0	1	0
2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0
12				10			
تصبح القيمة بلغة الـــ hexadecimal: **$(CA)_{16}$**							

المثال الثانى:

رقم $(00010111)_2$

0	0	0	1	0	1	1	1
2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0
1				7			
تصبح القيمة بلغة الـــ hexadecimal: **$(17)_{16}$**							

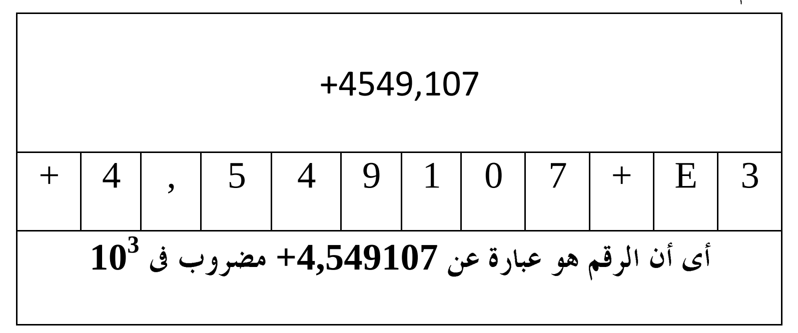
- **من الـــ real إلى الـــ floating point**:

تتم كتابة الإشارة لتوضيح أذا كان الرقم موجب أم سالب و بعد ذالك يكتب الرقم الأول ثم العلامة العشرية ثم تتم كتابة باقى الأرقم وبعد ذالك يكتب الأس لكى يعادل الرقم المكتوب بالـــ real.

١- رقم +4549,107
٢- رقم -719,17

المثال الأول:

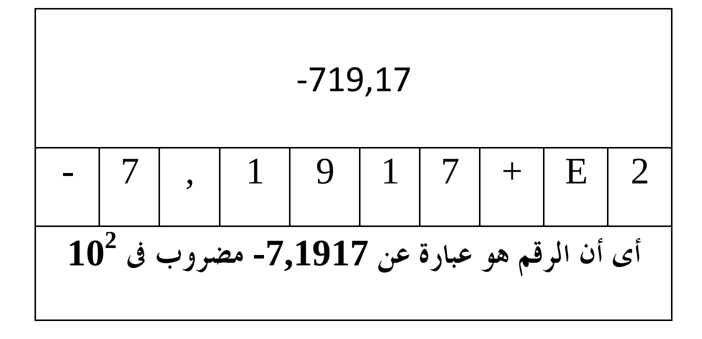
رقم +4549,107



+4549,107											
+	4	,	5	4	9	1	0	7	+	E	3
أى أن الرقم هو عبارة عن +4,549107 مضروب فى 10^3											

المثال الثانى:

رقم -719,17



-719,17									
-	7	,	1	9	1	7	+	E	2
أى أن الرقم هو عبارة عن -7,1917 مضروب فى 10^2									

- **من الـــ floating point إلى الـــ real**:

تتم كتابة الإشارة ثم الرقم ثم وضع العلامة العشرية حسب الأس ثم تتم كتابة باقى الرقم لكى يعادل الرقم المكتوب بالـــ floating point.

١- رقم +4,549107+E3

٢- رقم -6,11107+E3

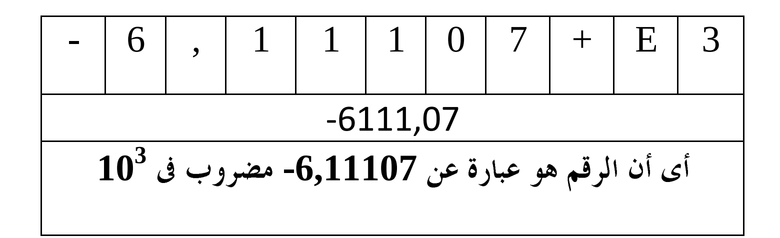
المثال الأول:

رقم +4,549107+E3

<table>
<tr><td>+</td><td>4</td><td>,</td><td>5</td><td>4</td><td>9</td><td>1</td><td>0</td><td>7</td><td>+</td><td>E</td><td>3</td></tr>
<tr><td colspan="12">+4549,107</td></tr>
<tr><td colspan="12">أى أن الرقم هو عبارة عن 4,549107+ مضروب فى 10^3</td></tr>
</table>

المثال الثانى:

رقم -6,11107+E3



<table>
<tr><td>-</td><td>6</td><td>,</td><td>1</td><td>1</td><td>1</td><td>0</td><td>7</td><td>+</td><td>E</td><td>3</td></tr>
<tr><td colspan="11">-6111,07</td></tr>
<tr><td colspan="11">أى أن الرقم هو عبارة عن 6,11107- مضروب فى 10^3</td></tr>
</table>

الأرقام الصحيحة........Integer Math :

تنقسم الأرقام الصحيحة إلى أرقام دون إشارة وأرقام بإشارة و تنقسم الأرقام بالإشارة إلى أرقام موجبة و أرقام سالبة.

- **أرقام بدون إشارة.........Unsigned**:

نظراً أنه لا توجد إشارة أذن نقوم باستخدام جميع الـــ bits.

فمثلاً:

- **أكبر رقم صحيح يمكن كتابته على الـــ byte هو**:

1	**1**	**1**	**1**	**1**	**1**	**1**	**1**
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
المجموع الكلى لمحتوى الـــ byte هو 255 فى هذه الحالة.							

- **أصغر رقم صحيح يمكن كتابتة على الـــ byte هو**:

0	**0**	**0**	**0**	**0**	**0**	**0**	**0**
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	0	0	0	0	0	0
المجموع الكلى لمحتوى الـــ byte هو صفر فى هذه الحالة.							

- **رقم 22 باستخدام ذاكرة byte بدون إشارة:**

0	**0**	**0**	**1**	**0**	**1**	**1**	**0**
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	0	16	0	4	2	0
المجموع الكلى لمحتوى الـــ byte هو 22 فى هذه الحالة.							

- **الأرقام ذات إشارة........Signed:**

نظراً أنه توجد إشارة أذن نقوم باستخدام جميع الـــ bits للأرقام ماعدا الـــ bit الأخير على اليسار لأنه يستخدم للإشارة.

تنقسم الأرقام ذات الإشارة إلى أرقام موجبة و أرقام سالبة:

أولاً. الأرقام الموجبة:

يرمز للإشارة الموجبة برقم صفر ويكتب فى الـــ bit الأخير على اليسار.

فمثلاً:

- **أكبر رقم موجب صحيح يمكن كتابته على الـــ byte هو:**

0	**1**	**1**	**1**	**1**	**1**	**1**	**1**
+	2^6	2^5	2^4	2^3	2^2	2^1	2^0
+	64	32	16	8	4	2	1
المجموع الكلى لمحتوى الـــ byte هو 127+ فى هذه الحالة.							

- **أصغر رقم موجب صحيح يمكن كتابته على الـــ byte هو:**

0	**0**	**0**	**0**	**0**	**0**	**0**	**0**
+	2^6	2^5	2^4	2^3	2^2	2^1	2^0
+	0	0	0	0	0	0	0
المجموع الكلى لمحتوى الـــ byte هو 0+ فى هذه الحالة.							

- **رقم 22+ باستخدام ذاكرة byte بدون إشارة:**

0	**0**	**0**	**1**	**0**	**1**	**1**	**0**
+	2^6	2^5	2^4	2^3	2^2	2^1	2^0
+	0	0	16	0	4	2	0
المجموع الكلى لمحتوى الـــ byte هو 22+ فى هذه الحالة.							

ثانياً. الأرقام السالبة:

يرمز للإشارة السالبة برقم واحد ويكتب فى الـــ bit الأخير على اليسار.

فمثلاً:

- **أكبر رقم سالب صحيح يمكن كتابته على الـــ byte** هو:

1	**1**	**1**	**1**	**1**	**1**	**1**	**1**
-2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
-128	64	32	16	8	4	2	1
المجموع الكلى لمحتوى الـــ byte هو 1- فى هذه الحالة.							

- **أصغر رقم سالب صحيح يمكن كتابته على الـــ byte** هو:

1	**0**	**0**	**0**	**0**	**0**	**0**	**0**
-2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
-128	0	0	0	0	0	0	0
المجموع الكلى لمحتوى الـــ byte هو 128- فى هذه الحالة.							

- **رقم 22– بأستخدام ذاكرة byte بدون إشارة**:

1	**1**	**1**	**0**	**1**	**0**	**1**	**0**
-2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
-128	64	32	0	8	0	2	0
المجموع الكلى لمحتوى الـــ byte هو 22- فى هذه الحالة.							

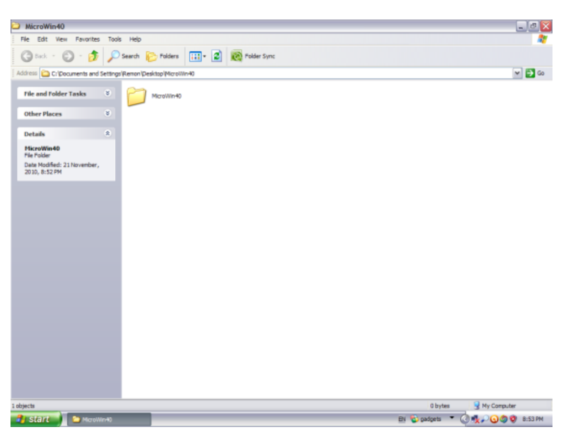
الباب الثالث

البرنامج

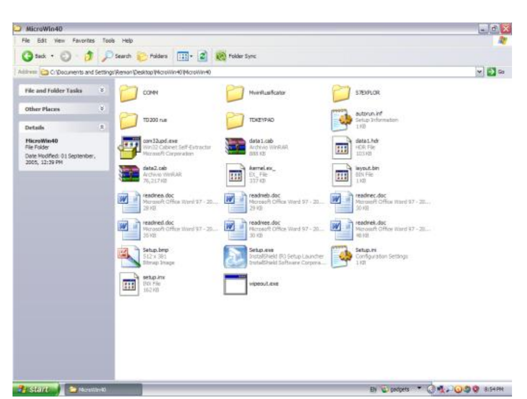
- كيفية تثبيت البرنامج.
- تعريف جهاز الـ PLC على الـ computer.
- تحديد المتغيرات الخاصة بالـ communication.
- تحديد المتغيرات الخاصة بالـ PG/PC interface.
- تحديد المتغيرات الخاصة بالـ acccess path.
- تحديد المتغيرات الخاصة بالـ PPI.
- تحديد المتغيرات الخاصة بالـ local connection.
- كيفية كشف الأخطاء قبل التشغيل.
- كيفية أيقاف جهاز الـ PLC.
- خطوات تحميل أى تمرين.
- كيفية تشغيل جهاز الـ PLC.

طريقة تثبيت البرنامج:

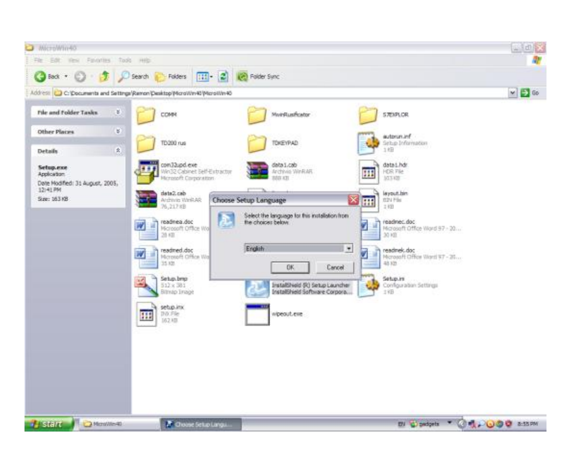
الملف الذى يحتوى البرنامج. ←



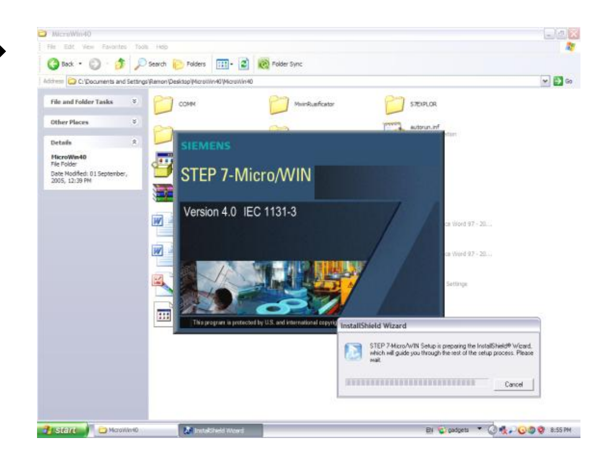
→ أضغط على ملف setup للتثبيت.



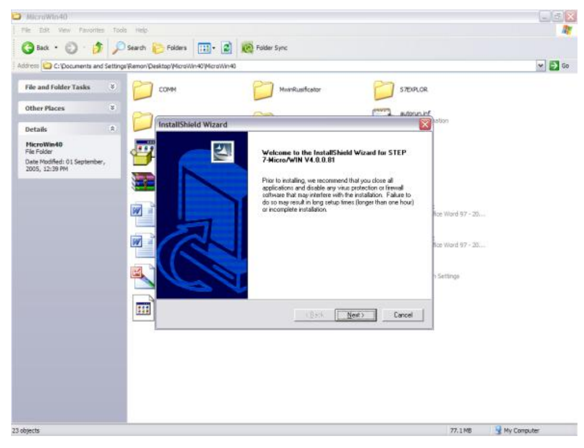
أختر اللغة المراد استخدمها. ←



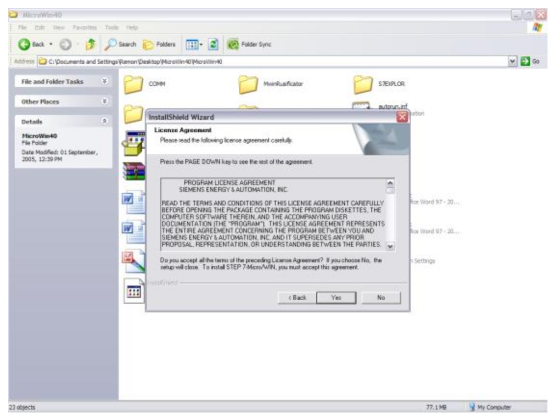
← يبدأ تثبيت البرنامج تدريجياً.



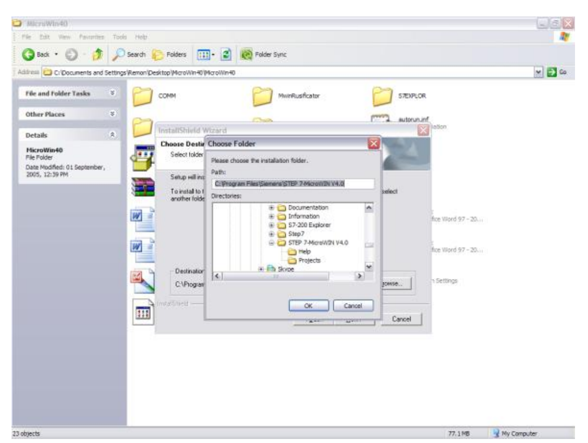
أضغط التالى next لإكمال التثبيت. →



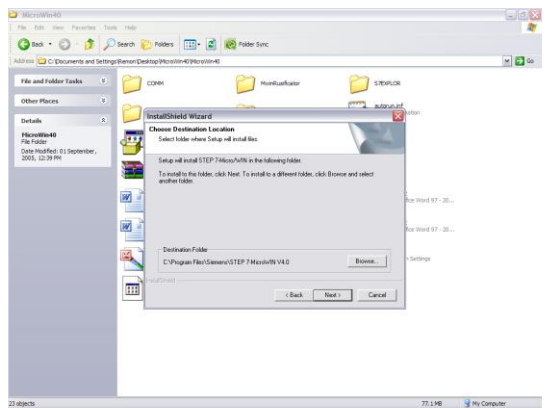
← أضغط نعم yes للموافقة على الشروط.



أضغط browse لاختيار مكان التثبيت. ←



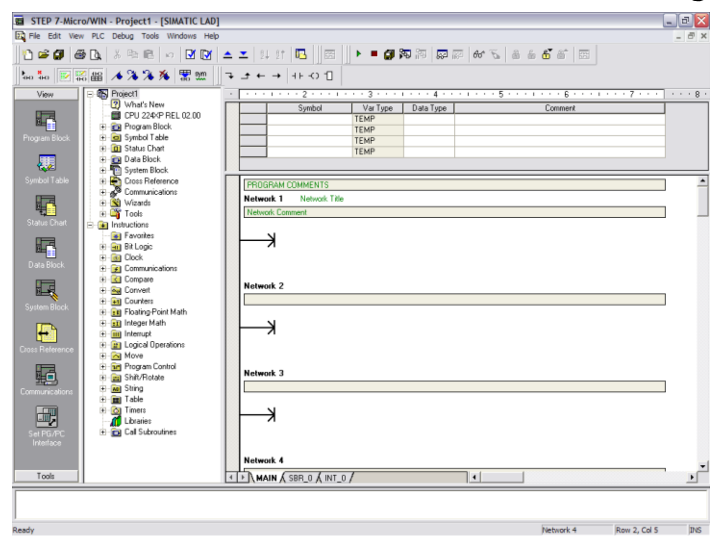
→ أضغط التالى next للأستمرار.



سوف يستمر التثبيت حتى النهاية. ←

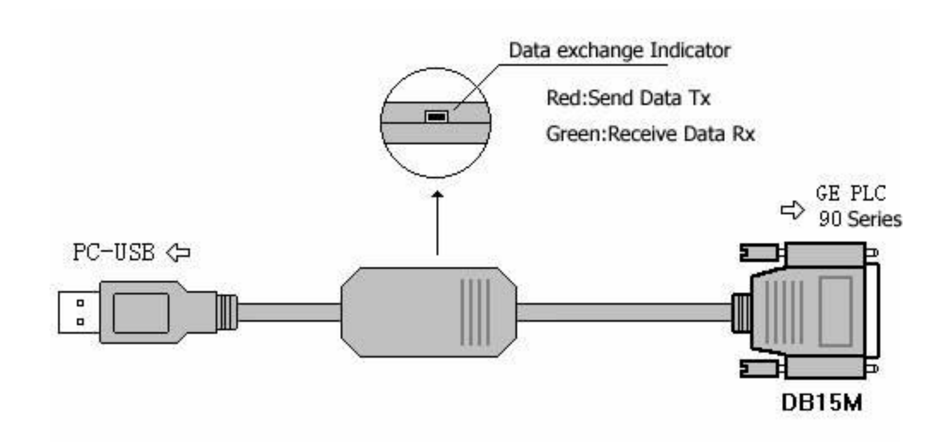


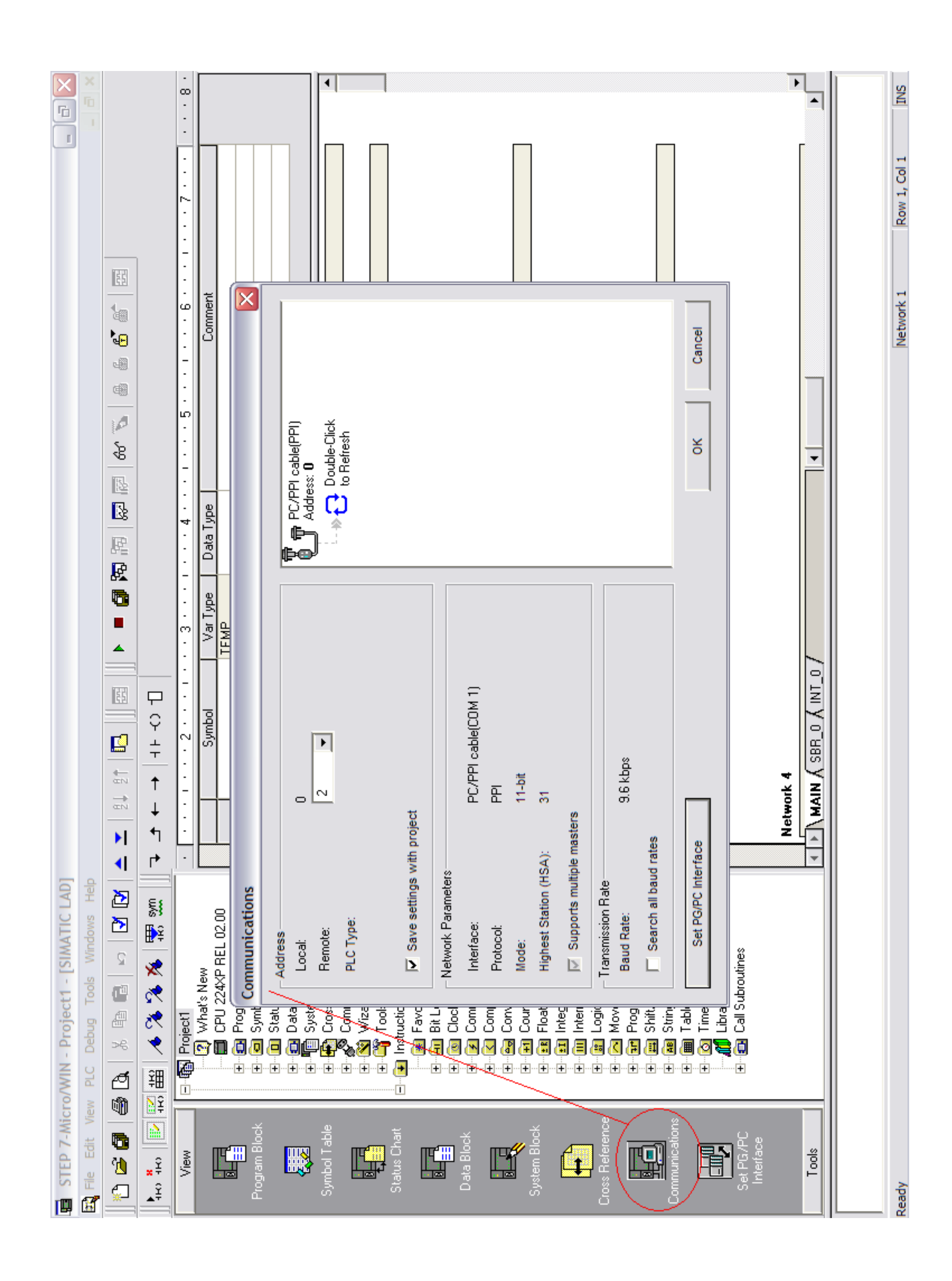
بعد تثبيت البرنامج. ←



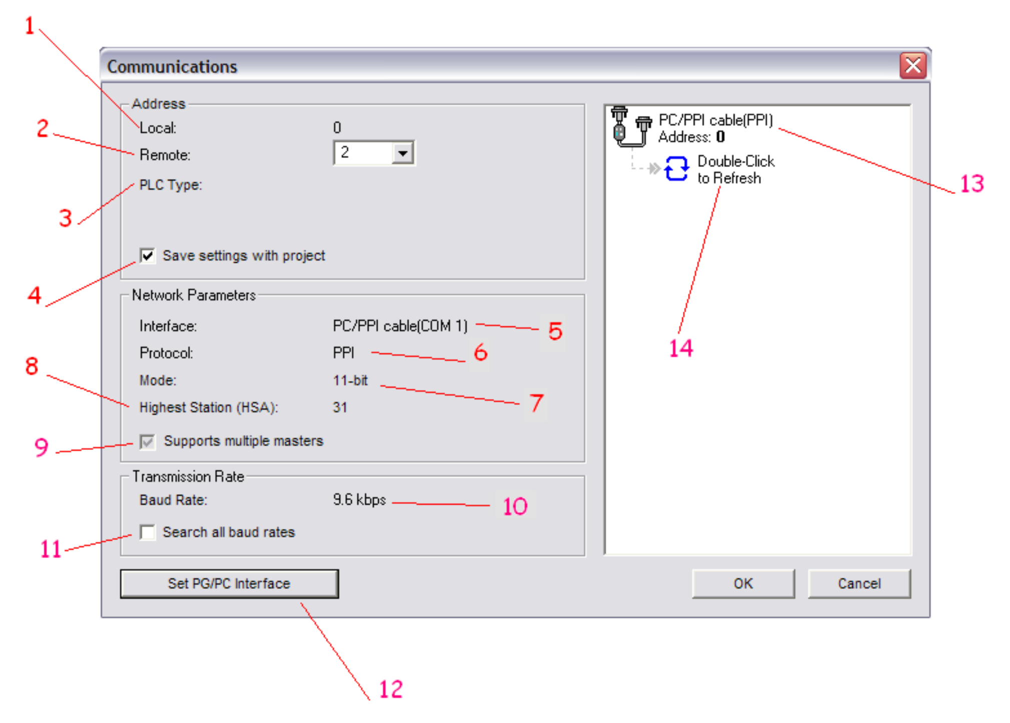
توصيل جهاز الــ PLC مع الكمبيوتر:

من المهم بعد تثبيت البرنامج الخاص بجهاز الــ PLC كما أوضح فى الخطوات السابقة أن يتم توصيل الكابل بحيث يتم توصيل الطرف الأول (Series) إلى جهاز الــ PLC و الطرف الأخر (USB) إلى الكمبيوتر وبعد ذالك يتم تعريف و تحديد بعض المتغيرات كما سوف يوضح بالصور الآتية:




STEP 7-Micro/WIN - Project1 - [SIMATIC LAD]
File Edit View PLC Debug Tools Windows Help
View
Program Block
Symbol Table
Status Chart
Data Block
System Block
Cross Reference
Communications
Set PG/PC Interface
Tools
Project1
What's New
CPU 224XP REL 02.00
Call Subroutines
Symbol
Var Type
Data Type
Comment
TEMP
Communications
Address
Local:
0
Remote:
2
PLC Type:
Save settings with project
Network Parameters
Interface:
PC/PPI cable(COM 1)
Protocol:
PPI
Mode:
11-bit
Highest Station (HSA):
31
Supports multiple masters
Transmission Rate
Baud Rate:
9.6 kbps
Search all baud rates
Set PG/PC Interface
PC/PPI cable(PPI)
Address: 0
Double-Click
to Refresh
OK
Cancel
Network 4
MAIN
SBR_0
INT_0
Ready
Network 1
Row 1, Col 1
INS

صفحة الـــ COMMUNICATIONS :



١- المقصود بـــ Local هو الكمبيوتر ويرمز له برقم صفر ولكن يمكن تغير الرقم.

٢- المقصود بـــ Remote هو الـــ PLC ويرمز له برقم أثنين ولكن يمكن تغير الرقم.

٣- المقصود بـــ PLC Type هو نوع الجهاز الموصل بالكابل.

٤- المقصود بـــ Save setting هو حفظ جميع المتغيرات مع البرنامج, مثلاً: لون و حجم الخط.

٥- المقصود بـــ Interface هو نوع الكابل المستخدم فالتوصيل.

٦- المقصود بـــ Protocol هو نوع النظام المتبع فالتوصيل من نقطة إلى نقطة.

٧- المقصود بـــ Mode هو نوع الـــ processor الخاص بالـــ CPU.

٨- المقصود بـــ Highest Station هو أقصى عدد من أجهزة الـــ PLC المراد التحكم بها بواسطة الكمبيوتر.

٩- المقصود بـــ Supports هو أن جهاز الـــ PLC موصل بأكثر من كمبيوتر أو HMI.

١٠- المقصود بـ Baud Rate هى سرعة تقل المعلومات من و إلى جهاز الـPLC.

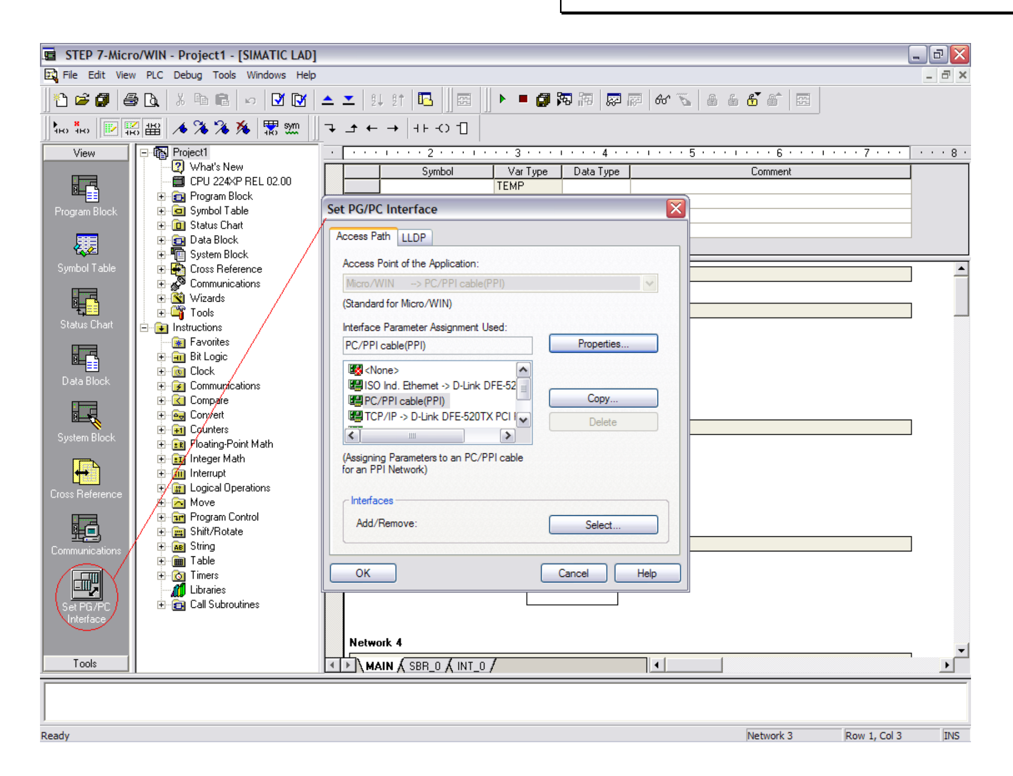
١١- المقصود بـ Search all baud rates لكى يختار سرعة تقل المعلومات من و إلى جهاز الـPLC أوتوماتيكياً.

١٢- المقصود بـ Set PG/PC هى صفحة لتحديد بعض المتغيرات الخاصة بالـ communication .

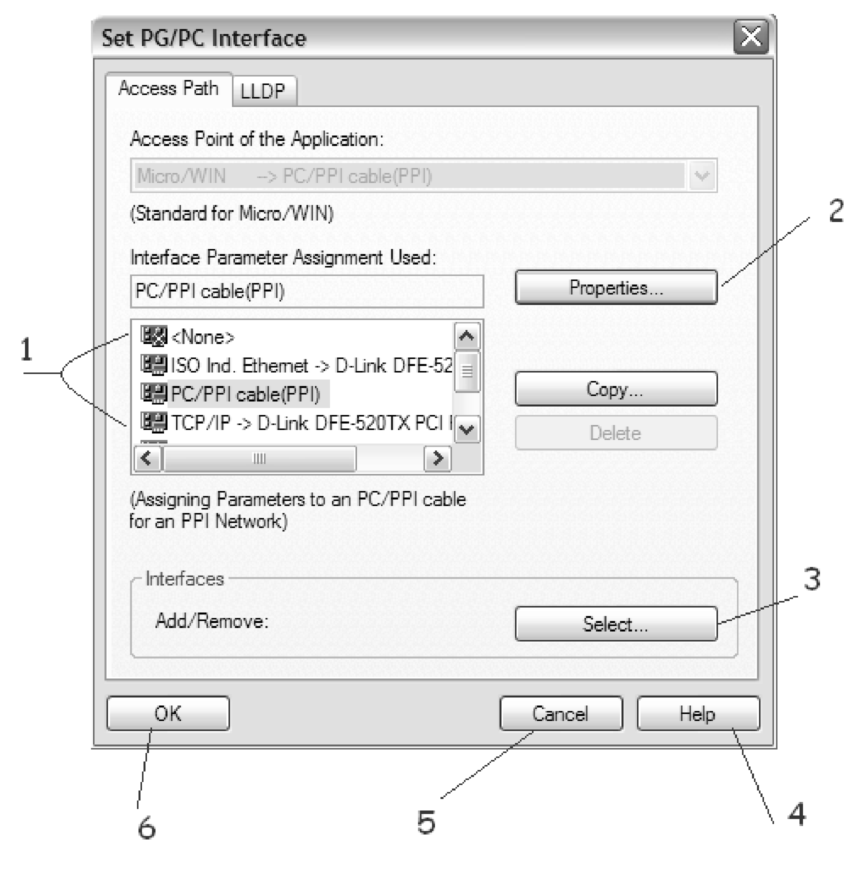
١٣- المقصود بـ PC/PPI cable يوضح رقم جهاز الـPLC و رقم الكمبيوتر فى حالة التواصل.

١٤- المقصود بـ Double click to refresh تستخدم فى حالة التوصيل لأول مرة للتوصيل.

صفحة Set PG/PC interface :



صفحة الـــ ACCCESS PATH :



١- طرق مختلفة للتحكم بجهاز الـــ PLC: (بواسطة الكابل أو شبكة النت أو .....).

٢ - المقصود بـــ Properties هى صفحة لتحديد بعض المتغيرات الخاصة بطريقة التحكم التى سوف يتم أختيارها.

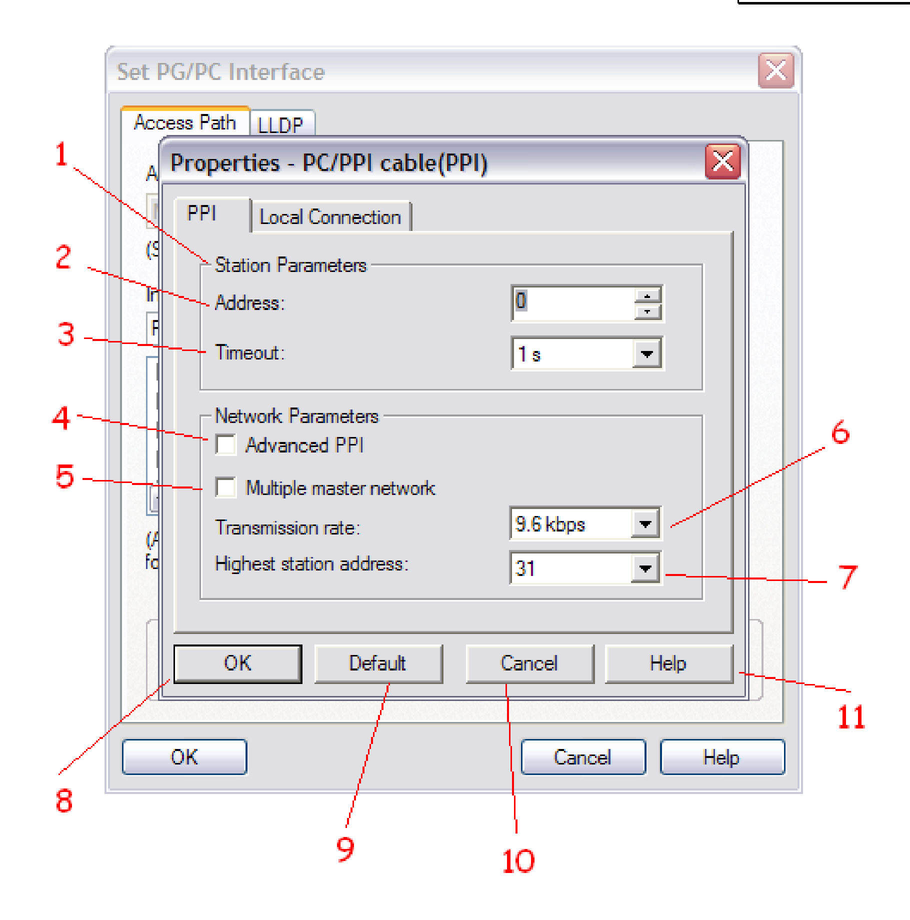
٣ - المقصود بـــ Select لاختيار الطرق المراد استخدامها فى التوصيل (كابل – نيت – شريحة محمول)

٤- المقصود بـــ Help هى صفحة تحتوى على شرح تفصيلى لصفحة الـــ SET PG/PC.

٥- المقصود بـــ Cancel تستخدم للخروج من الصفحة دون حفظ أى تغيير.

٦- المقصود بـــ Ok الخروج من الصفحة مع حفظ جميع التغيير.

صفحة الـــ PPI :



١- المقصود بـــ Station Parameters هي قأمة للمتغيرات الخاصة بوحدة الـــ PLC.

٢- المقصود بـــ Address هو العنوان الخاص بوحدة الـــ PLC.

٣- المقصود بـــ Timeout هو الوقت المحدد لظهور رسالة بعد إنقطاع الكابل.

٤- المقصود بـــ Advanced PPI هو التحكم بالـــ PLC باستخدام أكثر من مصدر.

٥- المقصود بـــ Multiple master network في حالة توصيل أكثر من جهاز PLC معاً.

٦- المقصود بـــ Transmission rate هي سرعة نقل المعلامات من و إلى الـــ PLC.

٧- المقصود بـــ Highest station address أقصى عدد من أجهزة الـــ PLC المراد التحكم بها بواسطة الكمبيوتر.

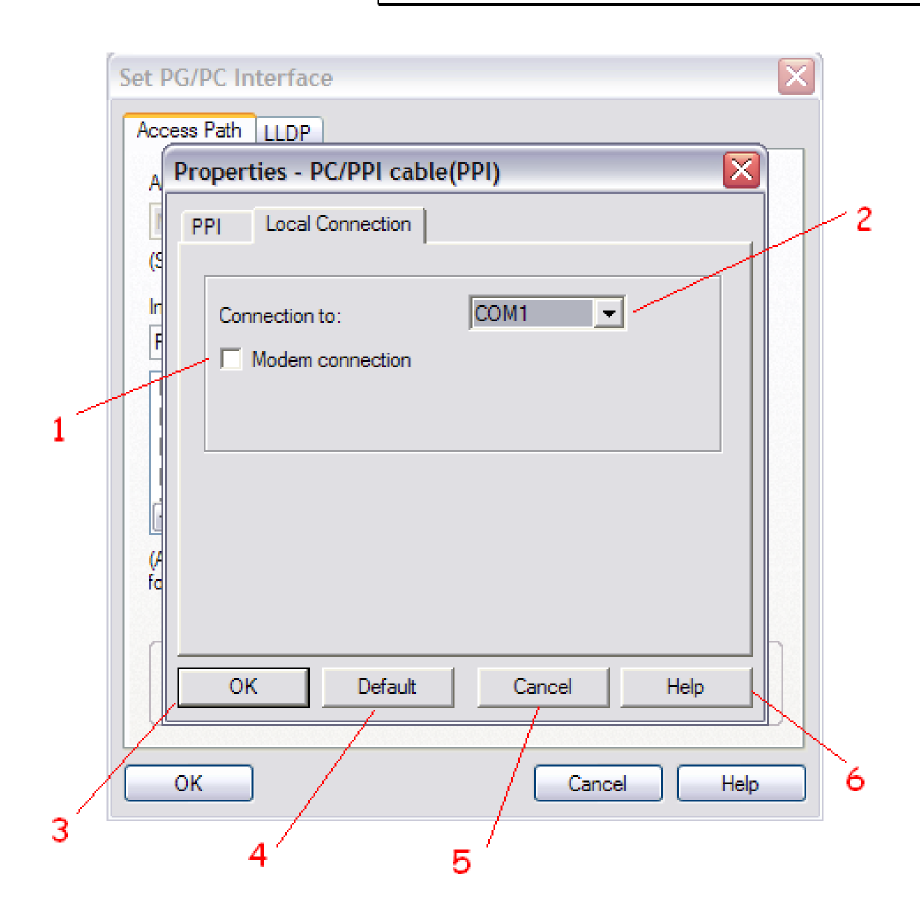
٨- المقصود بـــ Ok الخروج من الصفحة مع حفظ جميع التغيير.

٩- المقصود بـــ Default تستخدم للعودة المتغيرات إلى قيمتها الأصلية.

١٠- المقصود بـــ Cancel تستخدم للخروج من الصفحة دون حفظ أى تغيير.

١١- المقصود بـــ Help هى صفحة تحتوى على شرح شامل و تفصيلى لصفحة الـــ SET PG/PC.

صفحة الـــ LOCAL CONNECTION:



١- المقصود بـــ Connection to هو نوع طرف الكابل الموصل بالكمبيوتر. قد يكون: (USB أو COM)

٢- المقصود بـــ Modem connection هو التحكم بوحدة الـــ PLC بواسطة شبكة النت.

٣- المقصود بـــ Ok الخروج من الصفحة مع حفظ جميع التغيير.

٤- المقصود بـــ Default تستخدم للعودة المتغيرات إلى قيمتها الأصلية.

٥- المقصود بـــ Cancel تستخدم للخروج من الصفحة دون حفظ أى تغيير.

٦- المقصود بـــ Help هى صفحة تحتوى على شرح شامل و تفصيلى لصفحة الـــ SET PG/PC INTERFACE.

خطوات تحميل البرنامج:

١- يتم رسم البرنامج المراد تنفيذه كما سيوضح بالتفصيل فيما بعد.

PROGRAM COMMENTS
Network 1 Network Title
Network Comment
I0.1 I0.0 Q0.0
Q0.0
Network 2

٢- يتم الضغط على Compile all أو Compile لمعرفة أذا كانت توجد أخطاء فى الرموز المستخدمة.




```
Compiling Program Block...
MAIN (OB1)
SBR_0 (SBR0)
INT_0 (INT0)
Block Size = 24 (bytes), 0 errors

Compiling Data Block...
Block Size = 0 (bytes), 0 errors

Compiling System Block...
Compiled Block with 0 errors, 0 warnings

Total Errors: 0

Ready
```

ملاحظة: بالضغط على compile all أو compile:

- يظهر عدد الأخطاء كما فى الصورة السابقة بحيث يكتب Total Errors.
- فى حالة وجود أخطاء يكتب أيضاً مكان الرمز الخطاء و لماذا هو خطاء.
- بالضغط على الرسالة التى توضح الخطاء فأنه يظهر الرمز الخطاء تلقائيا.

٣- يتم الضغط على Stop للتأكد من أن الـــ PLC لا يعمل.



٤- يتم الضغط على Download لتحميل البرنامج من الـــ computer إلى الـــ PLC.

Download

PPI Connection 2

Use the Options button to select blocks to download. 3

1

Remote Address: 2 4

5 CPU 226 REL 01.21

Click Download to begin.

8

6 9 7 10 11 12

Options

Download

Cancel

Options

Program Block To : PLC

Data Block 13 To : PLC 14

System Block To : PLC

Recipes

Data Log Configurations

Click for Help and Support 15

Close dialog on success 16

Prompt on RUN to STOP 17

١- عنوان صفحة التحميل.

٢- طريقة التوصيل المستخدمة بين الـــ computer و الـــ PLC.

٣- وصف طريقة الاستخدام.

٤- عنوان الـــ CPU.

٥- نوع الـــ CPU.

٦- تعنى تنفيذ خطوات عملية التحميل دون أخطاء.

٧- المكان الذى يعرض فيه رسالة التحذير أو الخطاء.

٨- صورة متحركة لتوضيح عملية التحميل.

٩- يستخدم لإظهار أو أخفاء بعض الخيارات.

١٠-تظهر فقط فى حالة رسم البرنامج على CPU معينة وتحميله على CPU أخرى, ولكنها لا تظهر فى الصورة السابقة.

١١- بالضغط يبدأ عملية التحميل ولكن يجب أن لا تكون أى أخطاء.

١٢- لإلغاء عملية التحميل.

١٣- أختيار الأجزاء المراد تحميلها.

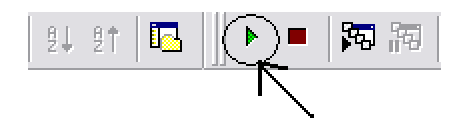
١٤- توضح أنه يتم نقل البرنامج إلى الـــ PLC.

١٥- للمساعدة بحيث أنه يفتح صفحة help.

١٦- يغلق صفحة التحميل تلقائياً عند أنتهاء التحميل بنجاح.

١٧- نظراً لأنه لا يمكن تحميل البرنامج بينما الـــ CPU فى وضع RUN فيقوم بتحويل الـــ CPU من RUN إلى STOP.

٥- يتم الضغط على Run للتأكد من أن جهاز الـــ PLC قد بدأ فى العمل وفى وقراءة البرنامج.



الباب الرابع

البرمجة

- لغـــــــــــــات البرمجة داخل جهاز الــ PLC.
- مــســميات الــمــدخــلات والــمــخــرجــات.
- لغــــــــــــــة المخطط السلمى (LAD).
- لغــــــــــــــة مخطط البوبات المنطقية (FBD).
- لغــــــــــــــة قائمة الأجراءات (STL).
- شـــــــــــــرح لغة المخطط السلمى بالتفصيل.
- شـــــــــــــرح الــ cycle time.
- شـــــــــــــرح الــ scan time.
- شــــرح كـيـفـيـة قــراءة الــبــرنــامــج.
- تمـــــــــــارين تطبيقية بأستخدام لغة ladder.

لغات البرمجة داخل جهاز الـــ PLC:

١- المخطط السلمى (Ladder Diagram Method):
٢- مخطط البوبات المنطقية (Function Block Diagram):
٣- قائمة الإجراءات (Statement List):

رقم	لغة البرمجة	اختصارها
١	Ladder Diagram Method	LAD
٢	Function Block Diagram	FBD
٣	Statement List	STL

- لغة الـــ LADDER هى أسهل اللغات فى البرمجة وذالك لأنها تشبه الكنترول كثيراً وهى التى سوف يتم التركيز عليها فى هذا الكتاب.

- لغة الـــ FBD هى ثانى أسهل اللغات فى البرمجة وذالك لأنها تشبه البوابات الإلكترونية كثيراً وسوف يتم شرح بعض التمارين بهذه اللغة فى الجزء الثانى من هذا الكتاب.

- لغة الـــ STL هى تعتبر أصعب اللغات فى البرمجة وذالك لأنها تتكون من كلمات ولكنها تتميز بحرية كتابة البرنامج دون تتبع ترتيب معين كما فى اللغتان الأخريتين وسوف يتم شرح بعض التمارين بهذه اللغة فى الجزء الثانى من هذا الكتاب.

مسميات المدخلات و المخرجات:

المدخلات:

مثال:

Ibit0.7 Ibit0.6 Ibit0.5 Ibit0.4 Ibit0.3 Ibit0.2 Ibit0.1 Ibit0.0

I Byte 0

مثال أخر:

Ibit1.7 Ibit1.6 Ibit1.5 Ibit1.4 Ibit1.3 Ibit1.2 Ibit1.1 Ibit1.0

I Byte 1

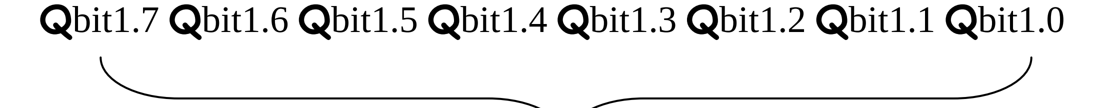
المخرجات:

مثال:

Qbit0.7 Qbit0.6 Qbit0.5 Qbit0.4 Qbit0.3 Qbit0.2 Qbit0.1 Qbit0.0

Q Byte 0

مثال أخر:



Q Byte 1

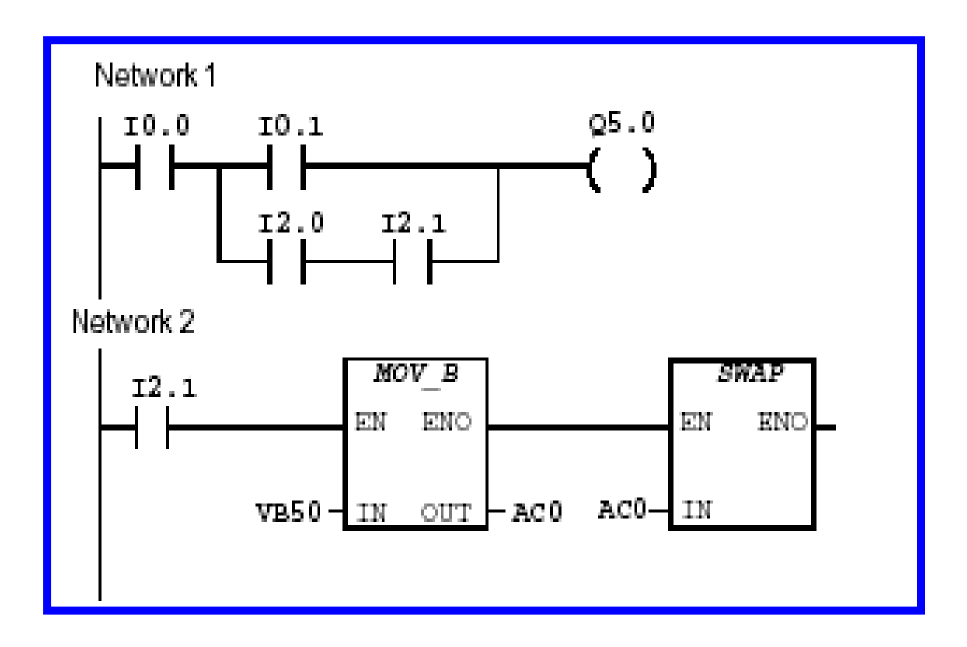
معلومات عن كل لغة:

المخطط السلمى (Ladder Diagram Method):

هذه الطريقة هى أقرب ما تكون لطريقة الكنترول المستخدم فى الدوائر الكهربية ودوائر التحكم و هى تعتبر من أسهل الطرق وأكثر الطرق استخداماً لعمل الدوائر الكهربية, مع ملاحظة أن الرسم يتم بطريقة أفقية.

رقم	مسمى الرمز	رمز الدائرة الكهربية	مسمى الرمز	رمز المخطط السلمى	الوظيفة
١	**S1**		**I0.3**		**مفتاح تشغيل**
٢	**S2**		**I1.7**		**مفتاح إيقاف**
٣	**H1**		**Q1.1**		**مخرجات**

- شكل عام للبرمجة بلغة الـ LADDER:

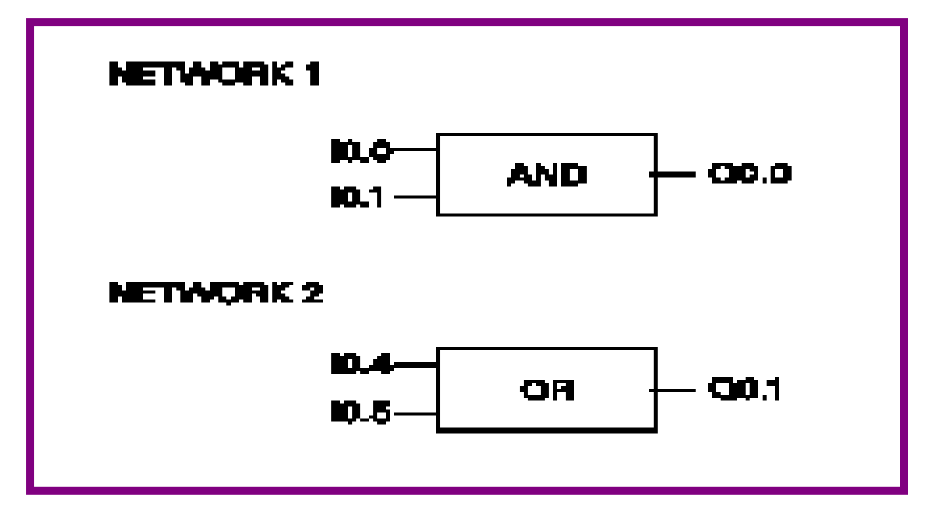


- **مخطط البوابات المنطقية (Function Block Diagram)**:

فى هذه الطريقة يتم استخدام البوابات المنطقية فى تنفيذ عمليات التحكم, وهى تعتبر من اللغات التى يسهل لمن يعمل فى مجال الإلكترونيات استخدامها, والبوابات المنطقية الأساسية الثلاث المستخدمة هى:

رقم	مسمى الرمز	رمز الدائرة الكهربية	مسمى الرمز	الوظيفة
١	أو	≥1	OR	مفتاحين على التوازى
٢	و	&	AND	مفتاحين على التوالى
٣	النفى		NOT	عكس الحالة

- شكل عام للبرمجة بلغة الـــ **FBD**:



- **قائمة الإجراءات (Statement List):**

هذة الطريقة يتم فيها وصف الدائرة المراد التحكم بها, بمجموعة أوامر, وهى مجموعة من الأوامر يعبر عنها بحروف.

رقم	مسمى الرمز	رمز الدائرة الكهربية	مسمى الرمز	الوظيفة
١	و	**A**	**AND**	مفتاح على التوالى
٢	أو	**O**	**OR**	مفتاح على التوازى
٣	لا	**N**	**NOT**	عكس الحالة

- **شكل عام للبرمجة بلغة الــ STL:**

```
NETWORK 1
        LD      I0.0
        A       I0.1
        =       Q0.0
NETWORK 2
        LD      I0.4
        O       I0.6
        =       Q0.1
```

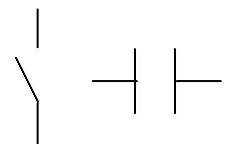
شرح لغة الــ Ladder Diagram Method:

نلاحظ أن ―| |― هو مفتاح يبدو مفتوحاً و يستخدم للتشغيل و فى نفس الوقت يبدو أن هذا المفتاح ―|/|― هو مفتاح مغلقاً و يستخدم للإيقاف ولكن هذا غير صحيح لأن لمعرفة أذا كان المفتاح مغلق أو مفتوح يعتمد هذا على نوع المفتاح الموصل على جهاز الــ PLC .

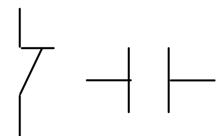
هذا المفتاح ―| |― له نفس حالة المفتاح الموصل على جهاز الــ PLC من الخارج.

هذا المفتاح ―|/|― له عكس حالة المفتاح الموصل على جهاز الــ PLC من الخارج.

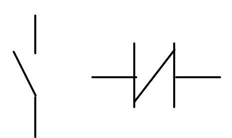
ملاحظة: هذا الشرح يخص فقط الـــ inputs وليس أى contact.



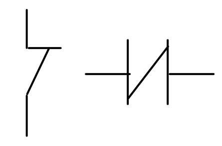
هو يعتبر مفتاح مفتوح (مفتاح تشغيل – normally open) لأنة له نفس حالة المفتاح الخارجى.



هو يعتبر مفتاح مغلق (مفتاح إيقاف – normally close) لأنة له نفس حالة المفتاح الخارجى.



هو يعتبر مفتاح مغلق (مفتاح إيقاف – normally close) لأنة له حالة عكس حالة المفتاح الخارجى.



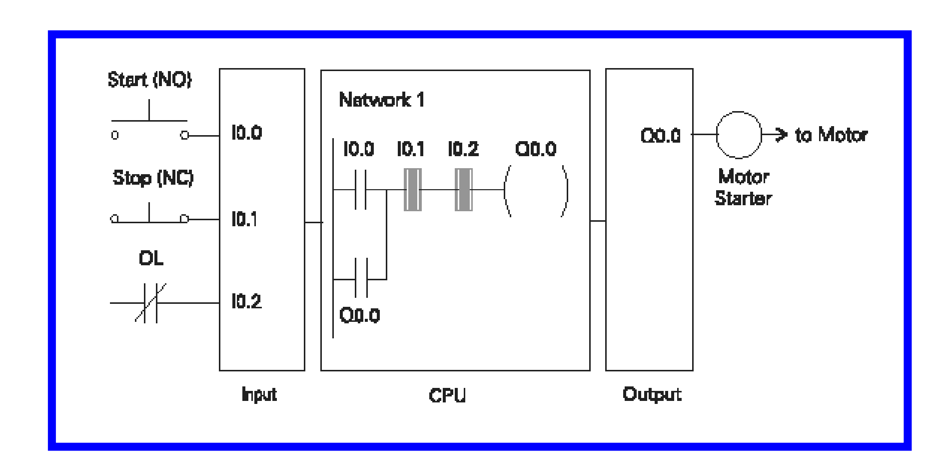
هو يعتبر مفتاح مفتوح (مفتاح تشغيل – normally open) لأنة له حالة عكس حالة المفتاح الخارجى.

مثال تجريبى: لمحرك يعمل من مكان واحد و يتوقف من مكان واحد مع وجود حماية overload.

شرح التمرين فى خمس خطوات:

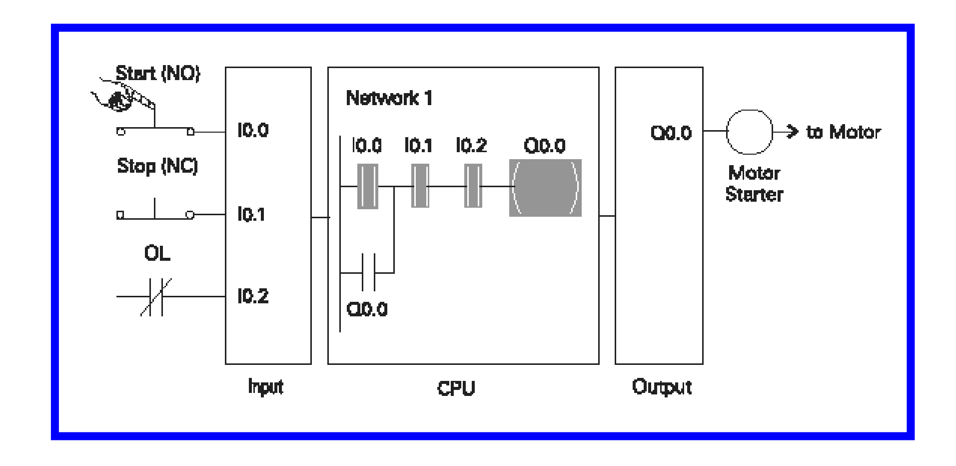
التمرين فى حالة إيقاف.

سوف نلاحظ أن I0.1 & I0.2 لهم نفس الحالة للمفاتيح التى بالخارج أى مغلقين.



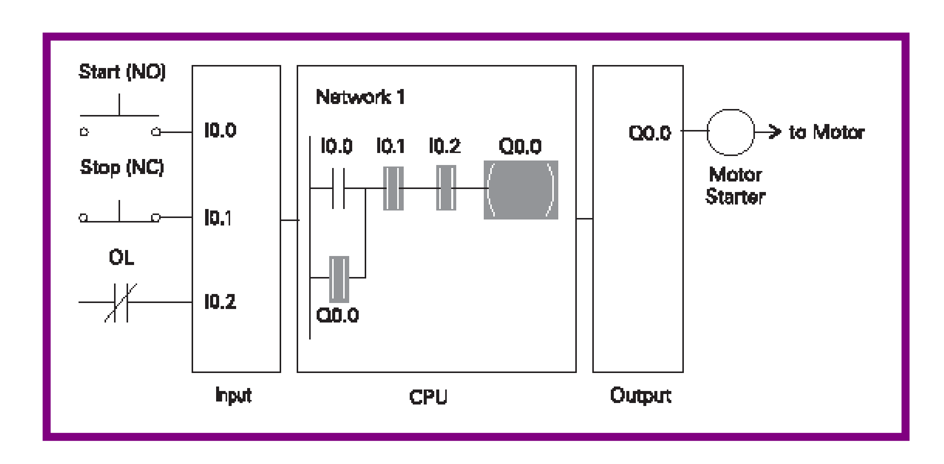
التمرين فى حالة تشغيل.

سوف نلاحظ أن I0.0 & I0.1 & I0.2 لهم نفس الحالة للمفاتيح التى بالخارج أى مغلقين.



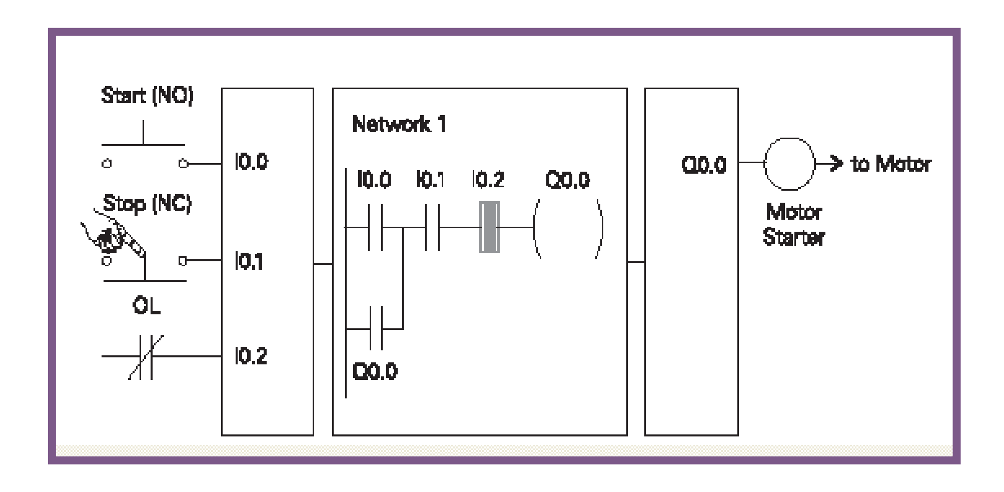
التمرين فى حالة تشغيل.

سوف نلاحظ أن نقطة التعويض أصبحت مغلقة.



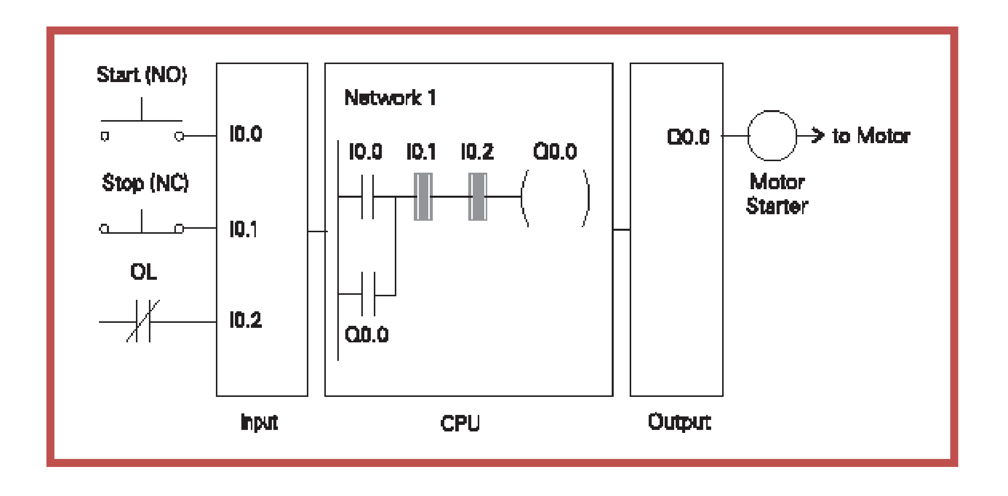
التمرين فى حالة إيقاف.

سوف نلاحظ أن I0.1 له نفس الحالة للمفتاح الذى بالخارج أى مفتوح.



التمرين فى حالة إيقاف.

سوف نلاحظ أن I0.1 & I0.2 لهم نفس الحالة للمفاتيح التى بالخارج أى مغلقين.

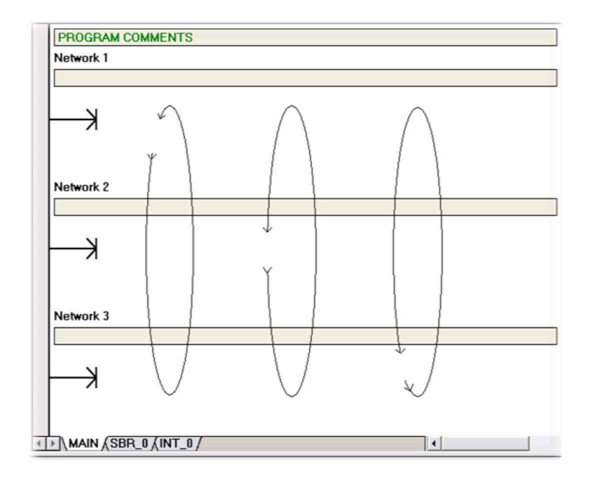


لمعرفة البرمجة فى الـــ **PLC** يجب معرفة:

- **أولاً:** Cycle time.
- **ثانياً:** Scan time.

الـــ Cycle time :

هو الـــزمن المستغـــرق لكـــي يقوم الـــ PLC بحلقة مغلقة كاملة أى بالتحرك من أى Network إلى أن يعـــود إلى نفس الـــ Network مرة أخـــرى ويتراوح الزمـــــــــن من 0.3ms إلى 1ms.



الـــ Scan time:

هو الوقت اللازم لوحدة الـــ PLC لكى:

١- يقراء المداخل (Read input).

٢- ينفذ البرنامج (Execute program).

٣- تعديل المخرجات (Update output).

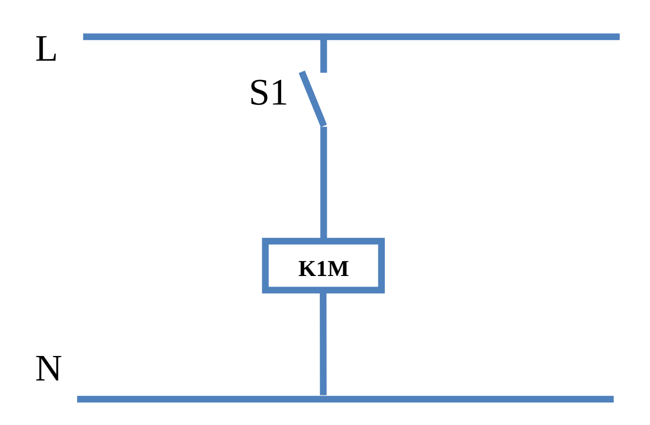
القانون: Scan time = Cycle time x Program size [Kbps]

ملاحظة: تتم قرأه البرنامج فى اتجاه واحد فقط من أعلى إلى أسفل و ليس العكس.

للتوضيح سوف نقوم ببعض الأمثلة:

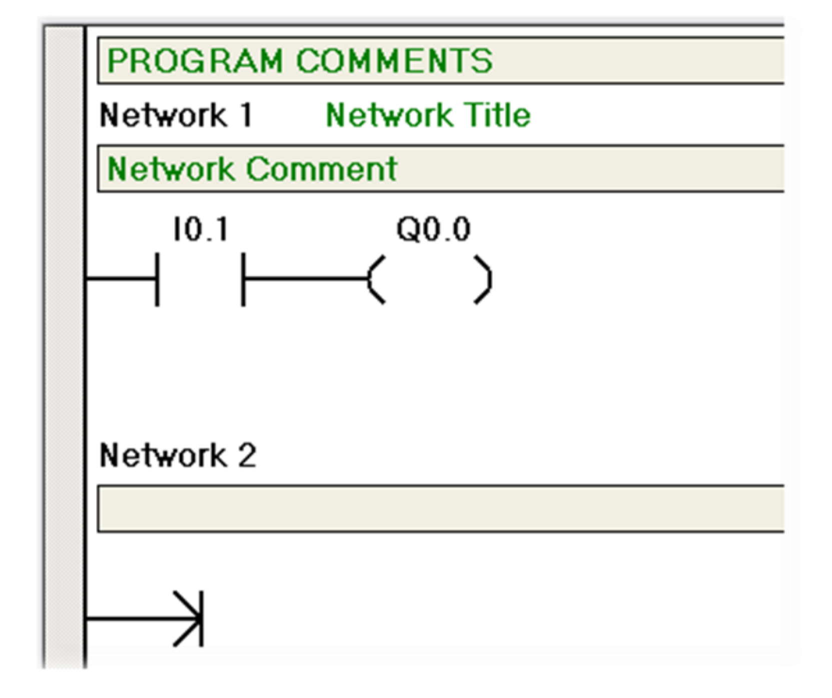
المثال الأول:

✓ محرك يعمل بواسطة مفتاح واحد:



أسم الدخل	نوع الدخل	عدد الدخل
I0.1/S1	n.o.	١
أسم الخرج	**نوع الخرج**	**عدد الخرج**
Q0.0/K1M	كونتاكتور	١

البرنامج:

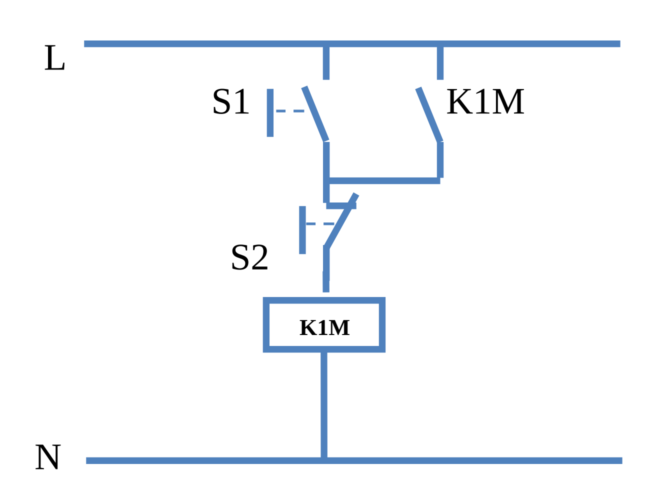


الشرح:

علماً بأن حالة المفتاح I0.1 بالخارج هو مفتوح فهو أيضاً كذلك فى البرنامج (أنظر صفحة 110) وبالضغط على المفتاح I0.1 بالخارج يغلق المفتاح أيضاً فى الداخل و يعمل الخرج Q0.0

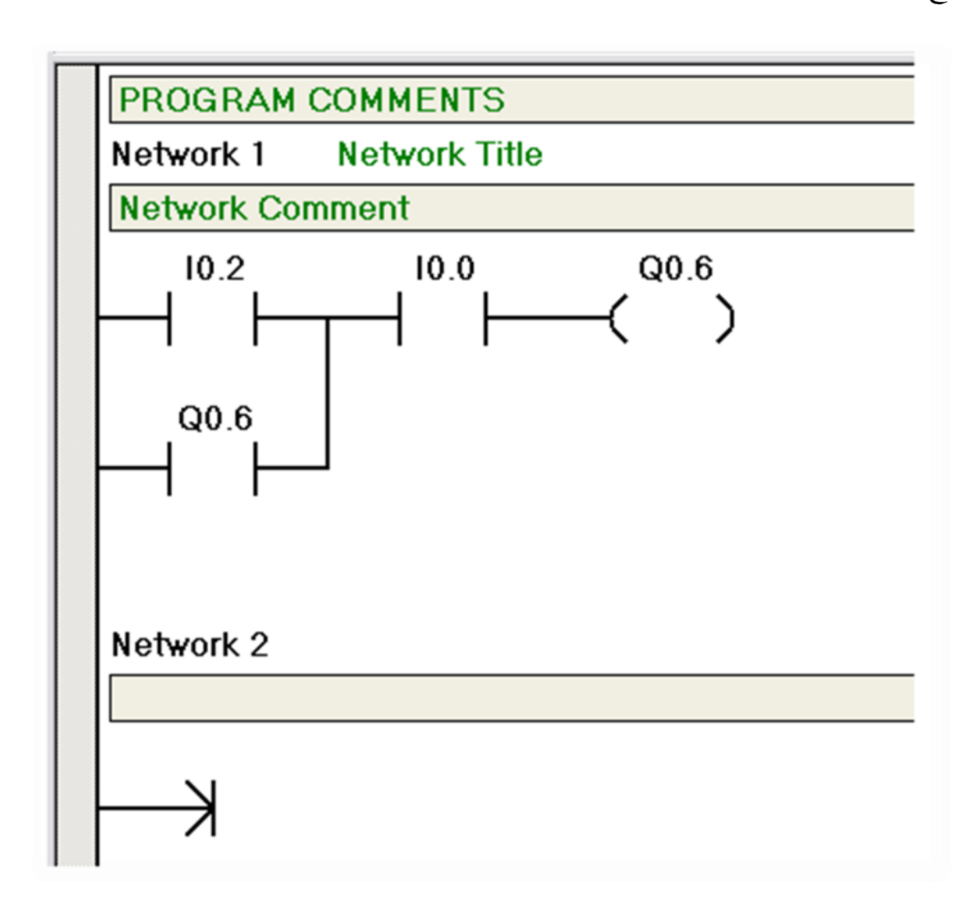
المثال الثانى:

✓ محرك يعمل بواسطة مفتاح واحد ويقف بواسطة مفتاح واحد:



عدد الدخل	نوع الدخل	أسم الدخل
١	n.o.	I0.2/S1
٢	n.c.	I0.0/S2
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتاكتور	Q0.6/K1M

البرنامج:



الشرح:

علماً بأن حالة المفتاح I0.1 بالخارج هو مفتوح فهو أيضاً كذلك فى البرنامج (أنظر صفحة 110).

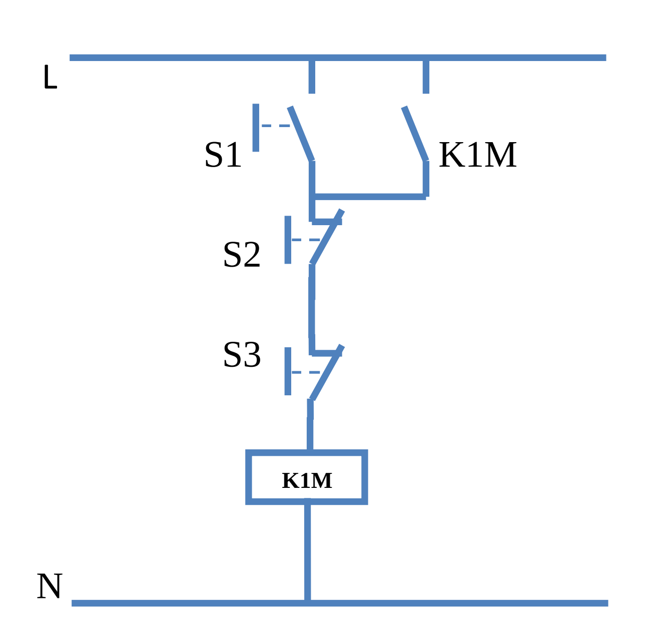
و حالة المفتاح I0.0 بالخارج هو مغلق فهو أيضاً كذلك فى البرنامج.

وبالضغط على المفتاح I0.1 بالخارج يغلق المفتاح أيضاً فى الداخل و يعمل الخرج Q0.0

وبالضغط على المفتاح I0.0 بالخارج يفتح المفتاح أيضاً فى الداخل و يقف الخرج Q0.0

المثال الثالث:

✓ محرك يعمل بواسطة مفتاح واحد ويقف بواسطة مفتاحين:



عدد الدخل	نوع الدخل	أسم الدخل
١	n.o.	I0.5/S1
٢	n.c.	I0.6/S2
٣	n.c.	I0.7/S3
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتاكتور	Q0.4/K1M

البرنامج:

PROGRAM COMMENTS

Network 1 Network Title

Network Comment

I0.5 I0.6 I0.7 Q0.4

Q0.4

Network 2

الشرح:

علماً بأن حالة المفتاح I0.5 بالخارج هو مفتوح فهو أيضاً كذلك فى البرنامج (أنظر صفحة 110).

و حالة المفتاح I0.6 بالخارج هو مغلق فهو أيضاً كذلك فى البرنامج.

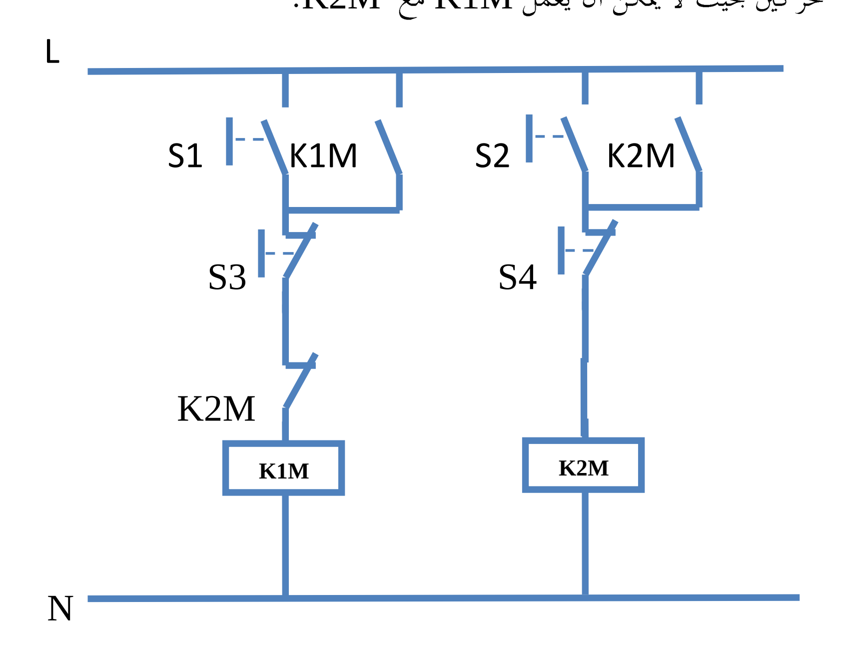
و حالة المفتاح I0.7 بالخارج هو مغلق فهو أيضاً كذلك فى البرنامج.

وبالضغط على المفتاح I0.5 بالخارج يغلق المفتاح أيضاً فى الداخل و يعمل الخرج Q0.0

وبالضغط على المفتاح I0.6 أو I0.7 بالخارج يفتح المفتاح أيضاً فى الداخل و يقف الخرج Q0.0

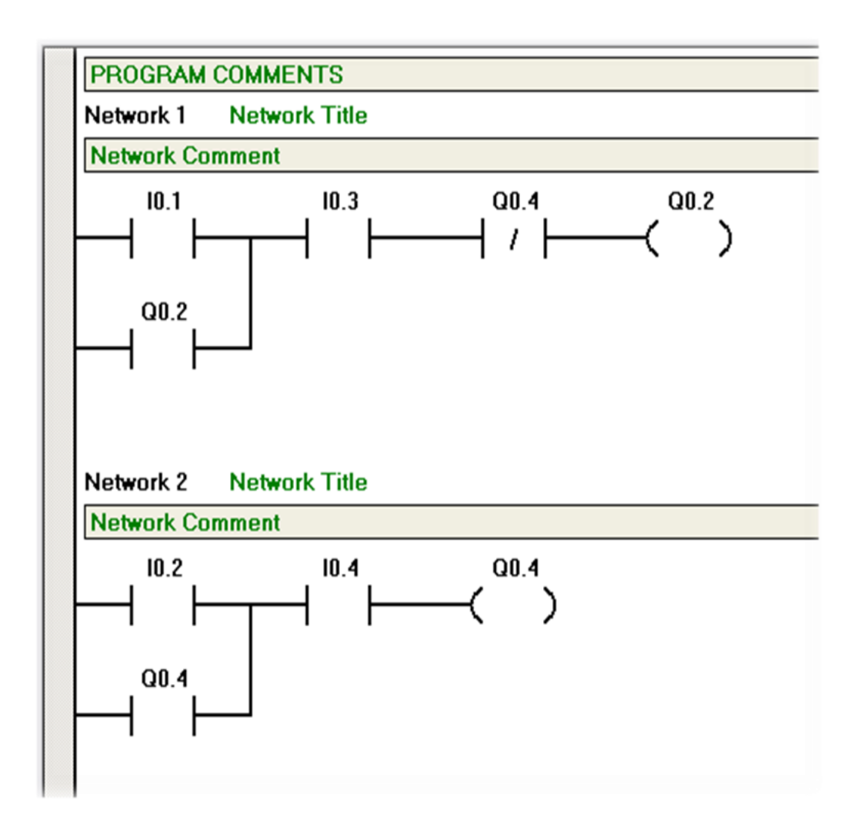
المثال الرابع:

✓ محركين بحيث لا يمكن أن يعمل K1M مع K2M:



أسم الدخل	نوع الدخل	عدد الدخل
I0.1 / S1	n.o.	١
I0.2 / S2	n.o.	٢
I0.3 / S3	n.c.	٣
I0.4 / S4	n.c.	٤
أسم الخرج	**نوع الخرج**	**عدد الخرج**
Q0.2 / K1M	كونتاكتور	١
Q0.4 / K2M	كونتاكتور	٢

البرنامج:



الشرح:

علماً بأن حالة المفتاح I0.1 و حالة المفتاح I0.2 بالخارج هو مفتوح فهو أيضاً كذلك فى البرنامج (أنظر صفحة 110).

وحالة المفتاح I0.3 و حالة المفتاح I0.4 بالخارج هو مغلق فهو أيضاً كذلك فى البرنامج.

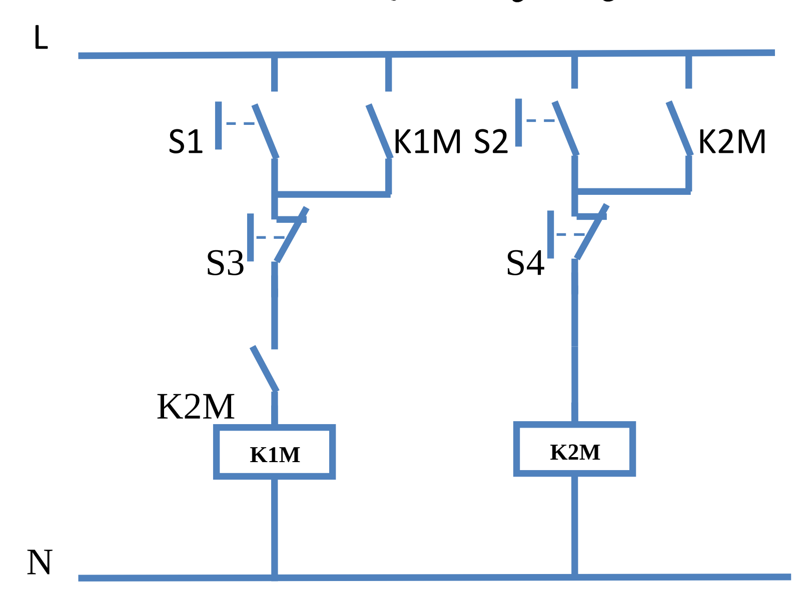
وبالضغط على المفتاح I0.1 بالخارج يغلق المفتاح أيضاً فى الداخل و يعمل الخرج Q0.2

وبالضغط على المفتاح I0.2 بالخارج يغلق المفتاح أيضاً فى الداخل و يعمل الخرج Q0.4

ولكن لا يمكن للإثنين أن يعملا معاً حيث أنه لا يمكن للخرج Q0.2 أن يعمل أثناء عمل الخرج Q0.4

المثال الخامس:

✓ محركين بحيث لا يمكن أن يعمل K1M دون K2M



عدد الدخل	نوع الدخل	أسم الدخل
١	n.o.	I0.1 / S1
٢	n.o.	I0.2 / S2
٣	n.c.	I0.3 / S3
٤	n.c.	I0.4 / S4
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتاكتور	Q0.2 / K1M
٢	كونتاكتور	Q0.4 / K2M

البرنامج:

PROGRAM COMMENTS

Network 1 Network Title

Network Comment

I0.1 I0.3 Q0.4 Q0.2

Q0.2

Network 2 Network Title

Network Comment

I0.2 I0.4 Q0.4

Q0.4

الشرح:

علماً بأن حالة المفتاح I0.1 و حالة المفتاح I0.2 بالخارج هو مفتوح فهو أيضاً كذلك فى البرنامج (أنظر صفحة 110).

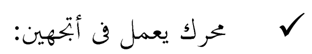
وحالة المفتاح I0.3 و حالة المفتاح I0.4 بالخارج هو مغلق فهو أيضاً كذلك فى البرنامج.

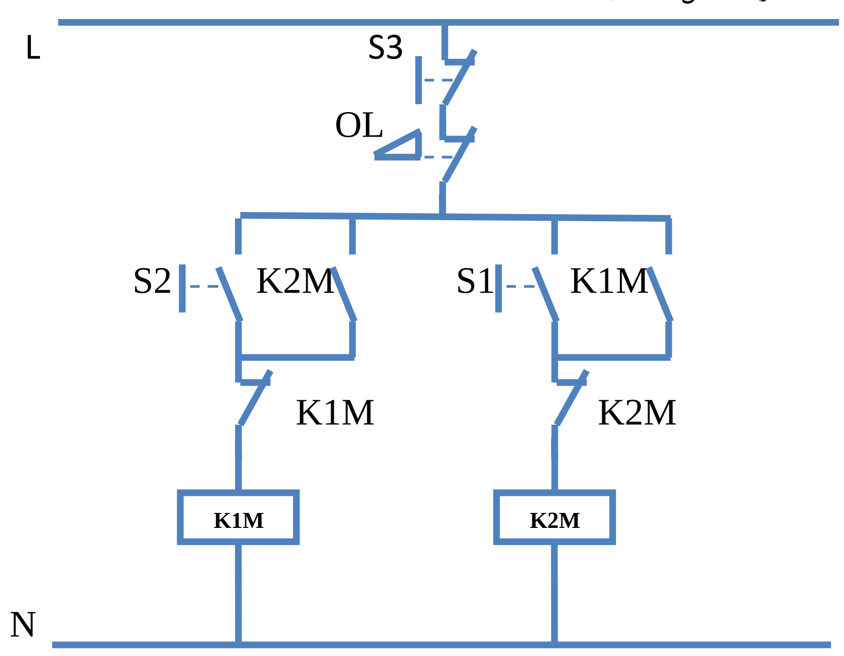
وبالضغط على المفتاح I0.1 بالخارج يغلق المفتاح أيضاً فى الداخل ولكن لا يمكن للخرج Q0.2 أن يعمل بسبب وجود نقطة مفتوحة من Q0.4

بينما بالضغط على المفتاح I0.2 بالخارج يغلق المفتاح أيضاً فى الداخل و يعمل الخرج Q0.4 دون الاعتماد على أى شروط أخرى

ولهذا يمكن للاثنين أن يعملا معاً حيث أنه يمكن للخرج Q0.2 أن يعمل أثناء عمل الخرج Q0.4 وليس بدونه.

المثال السادس:


محرك يعمل فى أتجهين:



أسم الدخل	نوع الدخل	عدد الدخل
I0.1 / S1	n.o.	١
I0.2 / S2	n.o.	٢
I1.1 / S3	n.c.	٣
I1.2 / OL	n.c.	٤
أسم الخرج	**نوع الخرج**	**عدد الخرج**
Q0.2 / K1M	كونتاكتور	١
Q0.4 / K2M	كونتاكتور	٢

البرنامج:

PROGRAM COMMENTS

Network 1 Network Title

Network Comment

I0.1 I1.1 I1.2 Q0.4 Q0.2

Q0.2

Network 2 Network Title

Network Comment

I0.2 I1.1 I1.2 Q0.2 Q0.4

Q0.4

الشرح:

علماً بأن حالة المفتاح I0.1 و حالة المفتاح I0.2 بالخارج هو مفتوح فهو أيضاً كذلك فى البرنامج (أنظر صفحة 110).

وحالة المفتاح I1.1 و حالة المفتاح I1.2 بالخارج هو مغلق فهو أيضاً كذلك فى البرنامج.

وبالضغط على المفتاح I0.1 بالخارج يغلق المفتاح أيضاً فى الداخل ويعمل الخرج Q0.2

وبالضغط على المفتاح I0.2 بالخارج يغلق المفتاح أيضاً فى الداخل ويعمل الخرج Q0.4

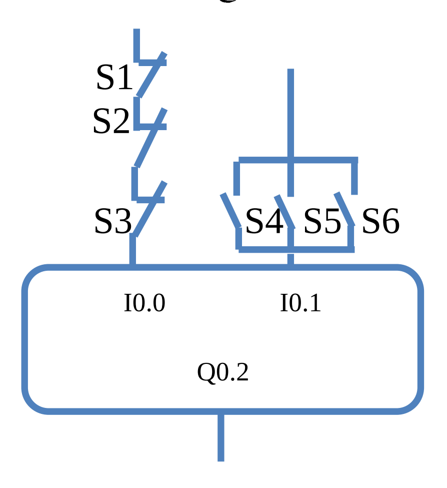
ولكن لا يمكن للاثنين أن يعملا معاً حيث أنه توجد نقطة مفتوح من كل خرج فى طريق الخرج الأخر ولهذا لا يمكن للخرج Q0.2 أن يعمل أثناء عمل الخرج Q0.4.

مثال أخر:

- ✓ محرك يعمل من ثلاث أماكن ويطفئ من ثلاث أماكن:

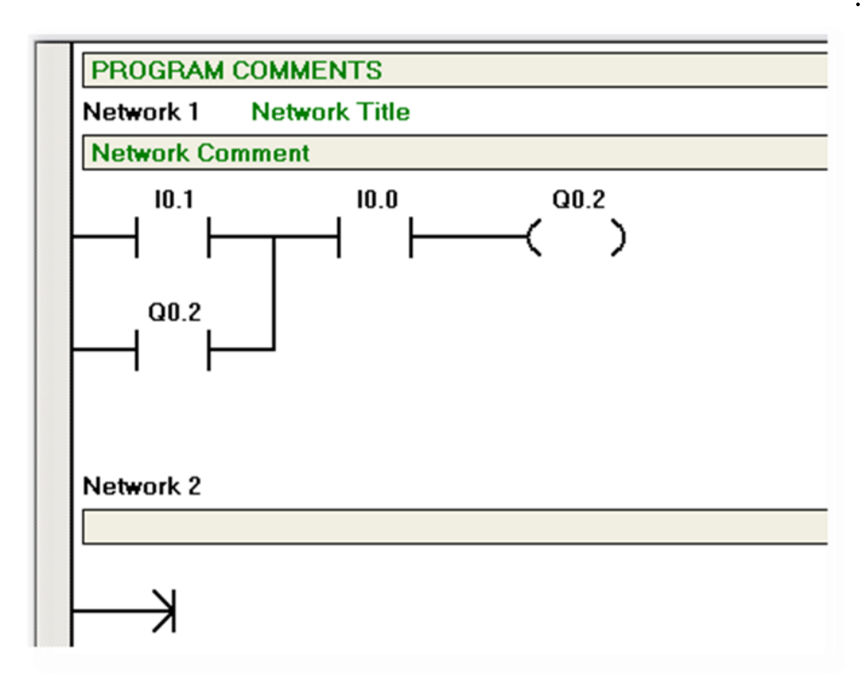
ملاحظة: بدلان من توصيل ستة مفاتيح على جهاز الــ PLC يمكن توصيل كل ثلاث مفاتيح على نقطة واحدة, سواء كان بالتوالى أو بالتوازى:

في هذه الحالة سيتم استخدام المفاتيح المغلقة للفصل والمفاتيح المفتوحة للتشغيل.



أسم الدخل	نوع الدخل	عدد الدخل
I0.0 / S1	n.c.	١
I0.0 / S2	n.c.	٢
I0.0 / S3	n.c.	٣
I0.1 / S4	n.o.	٤
I0.1 / S5	n.o.	٥
I0.1 / S6	n.o.	٦
أسم الخرج	**نوع الخرج**	**عدد الخرج**
Q0.2 / K1M	كونتاكتور	١

البرنامج:



الشرح:

علماً بأن حالة المفاتيح الموجودة على الدخل I0.1 فى الخارج هى مفاتيح مفتوحة وهى كذلك أيضاً فى البرنامج (أنظر صفحة 110).

وحالة المفاتيح الموجودة على الدخل I0.0 فى الخارج هى مفاتيح مغلقة وهى كذلك أيضاً فى البرنامج.

فبالضغط على أى من المفاتيح الموصلة بالدخل I0.1 يغلق المفتاح فى البرنامج و يعمل الخرج Q0.2

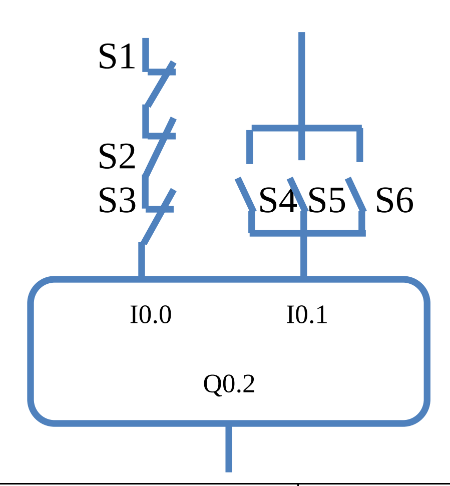
وبالضغط على أى من المفاتيح الموصلة بالدخل I0.0 يفتح المفتاح فى البرنامج و يقف الخرج Q0.2

مثال أخر:

✓ محرك يعمل من ثلاث أماكن ويطفأ من ثلاث أماكن:

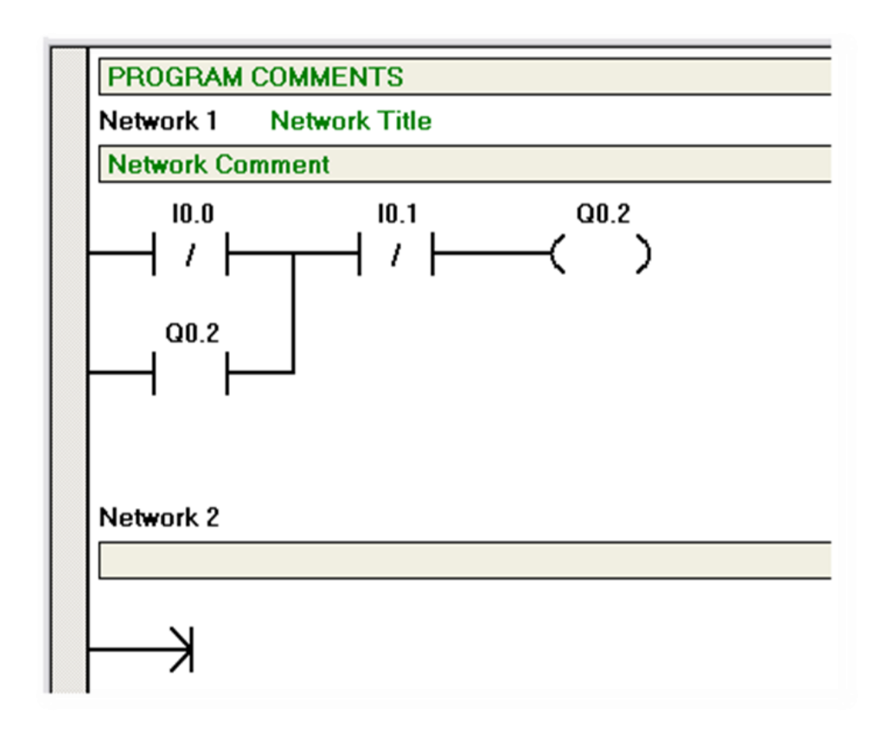
ملاحظة: بدلان من توصيل ستة مفاتيح على جهاز الـــ PLC يمكن توصيل كل ثلاث مفاتيح على نقطة واحدة, سواء كان بالتوالى أو بالتوازى:

في هذه الحالة سيتم استخدام المفاتيح المغلقة للتشغيل والمفاتيح المفتوحة للفصل.



أسم الدخل	نوع الدخل	عدد الدخل
I0.0 / S1	n.o.	١
I0.0 / S2	n.o.	٢
I0.0 / S3	n.o.	٣
I0.1 / S4	n.c.	٤
I0.1 / S5	n.c.	٥
I0.1 / S6	n.c.	٦
أسم الخرج	**نوع الخرج**	**عدد الخرج**
Q0.2 / K1M	كونتاكتور	١

البرنامج:



الشرح:

علماً بأن حالة المفاتيح الموجودة على الدخل I0.1 فى الخارج هى مفاتيح مفتوحة ولكنها مغلقة فى البرنامج بسبب استخدام مفتاح عكس الحالة (أنظر صفحة 110).

وحالة المفاتيح الموجودة على الدخل I0.0 فى الخارج هى مفاتيح مغلقة ولكنها مفتوحة فى البرنامج بسبب استخدام مفتاح عكس الحالة.

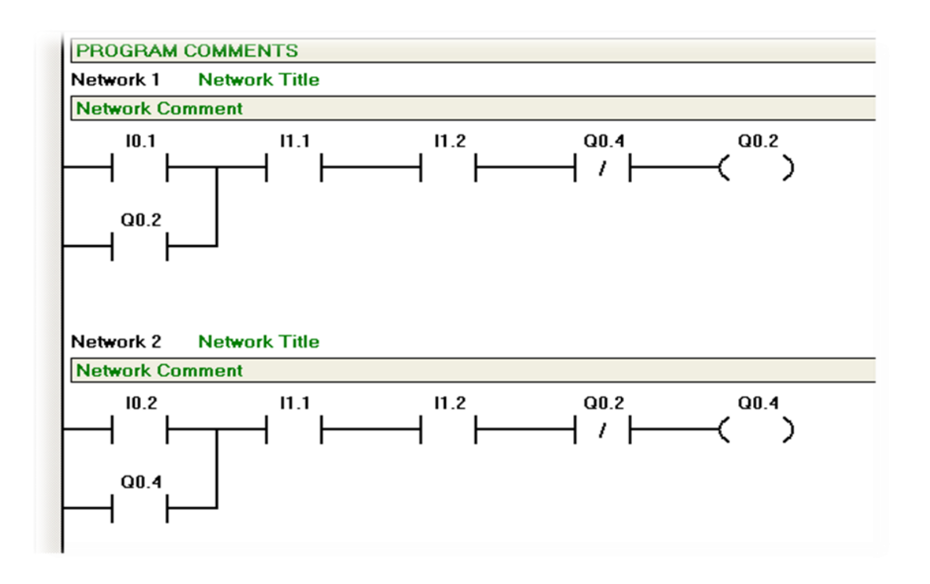
فبالضغط على أى من المفاتيح الموصلة بالدخل I0.0 تقطع الإشارة فى الخارج ولكن يغلق المفتاح فى البرنامج بسبب عكس الحالة و يعمل الخرج Q0.2

وبالضغط على أى من المفاتيح الموصلة بالدخل I0.1 ترسل إشارة إلى وحدة الـــ PLC من الخارج ولكن يفتح المفتاح فى البرنامج بسبب عكس الحالة و يقف الخرج Q0.2

لمعرفة طريقة قراءة الــ PLC لأى برنامج فلنستخدم كمثال للشرح, التمرين الذى رسمناه فى الصفحات السابقة وهو محرك اتجاهين.

عدد الدخل	نوع الدخل	أسم الدخل
١	n.o.	I0.1 / S1
٢	n.o.	I0.2 / S2
٣	n.c.	I1.1 / S3
٤	n.c.	I1.2 / S4

البرنامج:



في حالة الضغط على المفتاح I0.2:

رقم الـــ CYCLE	Network	ماذا يحدث
Cycle رقم (n)	Network1	لا يحدث أى شيء لأن Q0.4 لم تعمل بعد.
	Network2	تصبح جميع النقط مغلقة وتعمل Q0.4 وتغلق أيضاً نقطة التعويض.
Cycle رقم (n+1)	Network1	سوف تفتح النقطة المغلقة لـــ Q0.4.
	Network2	تبقى تعمل كما هى

في حالة فتح المفتاح I0.2:

رقم الـــ CYCLE	Network	ماذا يحدث
Cycle رقم (n+3000)	Network1	سوف تبقى نقطة Q0.4 مفتوحة.
	Network2	تبقى Q0.4 تعمل لأن نقطة التعويض مغلقة.
Cycle رقم (n+3001)	Network1	سوف تبقى نقطة Q0.4 مفتوحة.
	Network2	تبقى تعمل كما هى

كيف سوف يعمل هذا التمرين إذا قمنا:

١. بفتح مفتاح الإيقاف الخاص بالاتجاهين I1.1
٢. بغلق مفاتيح التشغيل I0.1 و I0.2
٣. ثم نقوم بعد ذلك بغلق مفتاح الإيقاف الخاص بالاتجاهين I1.1

الإجابة:

نظراً إلى أن وحدة الـــ PLC تقرأ البرنامج بنظام وترتيب فأن الوحدة ستمر أولاً على Q0.2 ثم تمر على Q0.4 ولهذا سوف تعمل Q0.2 ولهذا لا يمكن للخرج Q0.4أن يعمل فى هذه الحالة بسبب وجود نقطة مغلقة من الخرج الأول Q0.2 فلى طريق الخرج الثانى Q0.4.

الريليه...... Markers:

الـــ marker هو ريليه يستخدم داخل وحدة الـــ PLC فى البرمجة.

المسميات الخاصة بالـــ markers هى:

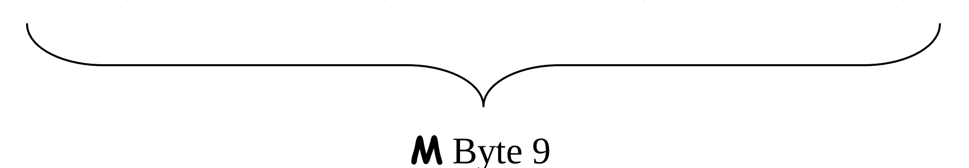
مثال:

Mbit0.7 **M**bit0.6 **M**bit0.5 **M**bit0.4 **M**bit0.3 **M**bit0.2 **M**bit0.1 **M**bit0.0



مثال أخر:

Mbit9.7 **M**bit9.6 **M**bit9.5 **M**bit9.4 **M**bit9.3 **M**bit9.2 **M**bit9.1 **M**bit9.0



الشكل:

M0.4

—( )

مثال:

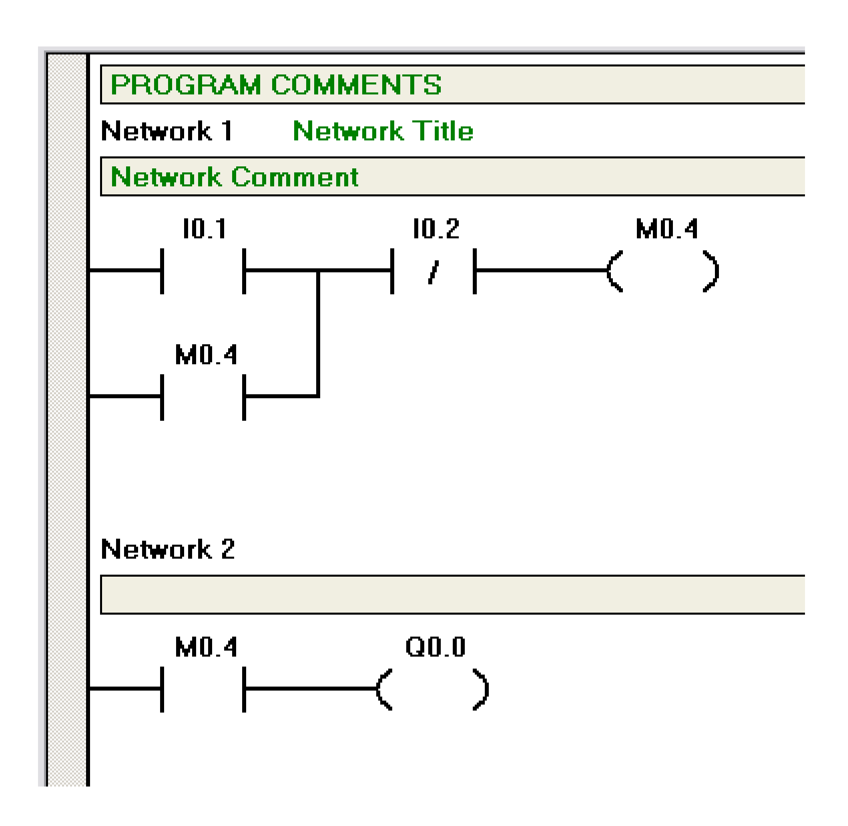
✓ محرك يعمل من مكان واحد و يقف من مكان واحد.

عدد الدخل	نوع الدخل	أسم الدخل
١	n.o.	I0.1/S1
٢	n.o.	I0.2/S2
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتاكتور	Q0.0/K1M

ملاحظة:

١- تم عكس حالة المفتاح I0.2 لأنة مفتوح بالخارج.
٢- الــ marker يستخدم كمساعد داخل الــ PLC ولا يمكن استخدامه كخرج.
٣- استخدام الــ marker فى هذا التمرين ليس عملياً ولكن تم استخدامه فقط للتوضيح.
٤- يمكن استخدام الــ marker مع الــ Set/Reset أيضاً كما سيتم شرحها بعد قليل.

البرنامج:



الشرح:

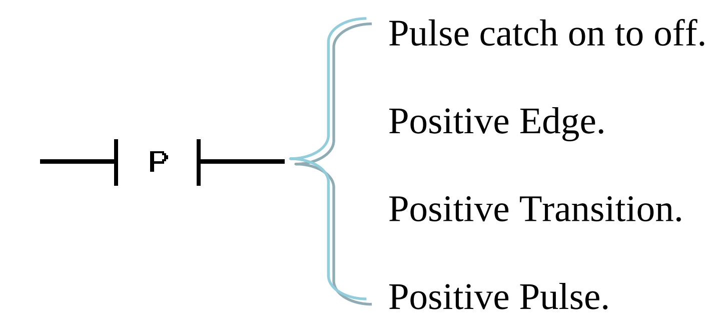
Network1:

بالضغط على I0.1 يعمل M0.4 وتغلق نقطة الحفظ بحيث فى حالة فتح I0.1 يستمر الـــ marker فى العمل.

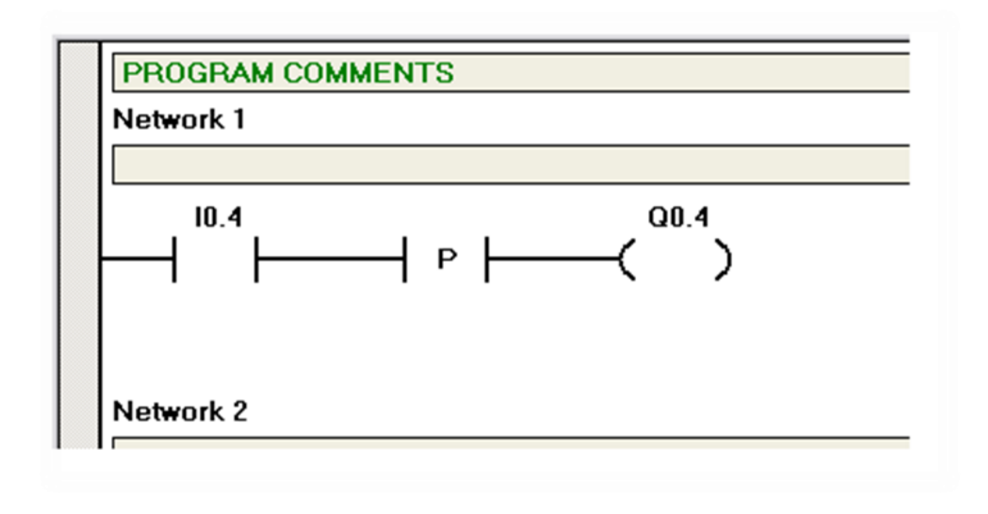
Network2:

بعد أن يعمل M0.4 يعمل فى نفس الـــ Cycle أيضاً الخرج Q0.0

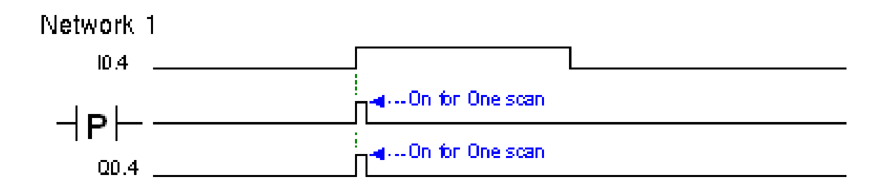
مفتاح الـــ Positive edge :



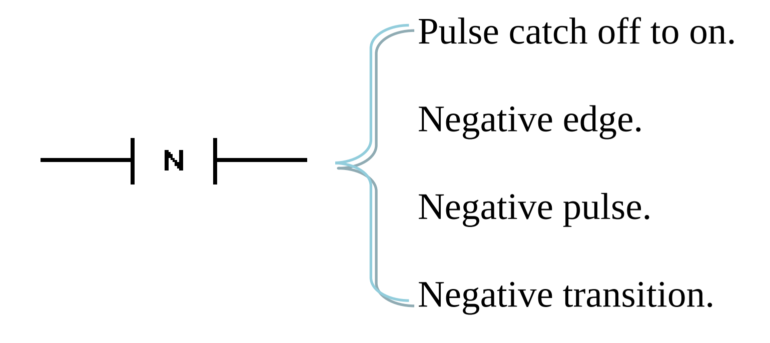
كل هذه المسميات السابقة هى لنفس الرمز.
مفتاح الـــ positive edge هو مفتاح يوصل بالتوالى بعد أى مفتاح أخر بحيث عندما نقوم بغلق المفاتيح التى تسبق مفتاح الـــ positive edge فيوصل المفتاح الإشارة لزمن يعادل زمن الـــ cycle time ولتكرير هذه الإشارة يجب فتح أى مفتاح من الذين يسبقوا الـــ positive edge ثم غلقه مرة أخرى.



رسم تخطيطى:



مفتاح الــ Negative edge :



كل هذه المسميات السابقة هى لنفس الرمز.

مفتاح الــ Negative edge هو مفتاح يوصل بالتوالى بعد أى مفتاح أخر بحيث عندما نقوم بغلق المفاتيح التى تسبق مفتاح الــ Negative edge فلا يوصل المفتاح الإشارة ولكن عند فتح أى أو كل المفاتيح التى تسبق مفتاح الــ Negative edge فيوصل المفتاح الإشارة لزمن يعادل زمن الــ cycle time ولتكرير هذه الإشارة يجب غلق أى مفتاح من الذين يسبقوا الــ Negative edge ثم فتحة مرة أخرى.

البرنامج

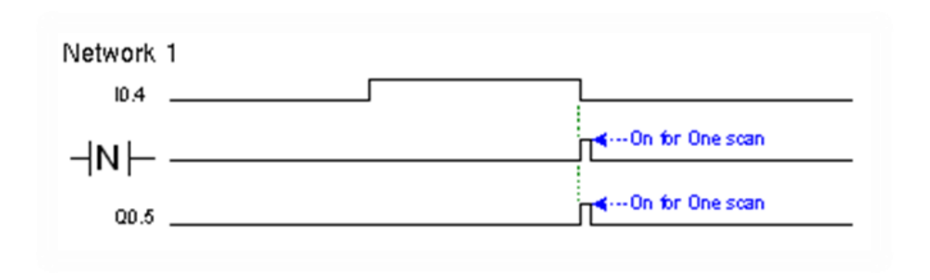
PROGRAM COMMENTS

Network 1

I0.4 N Q0.5

Network 2

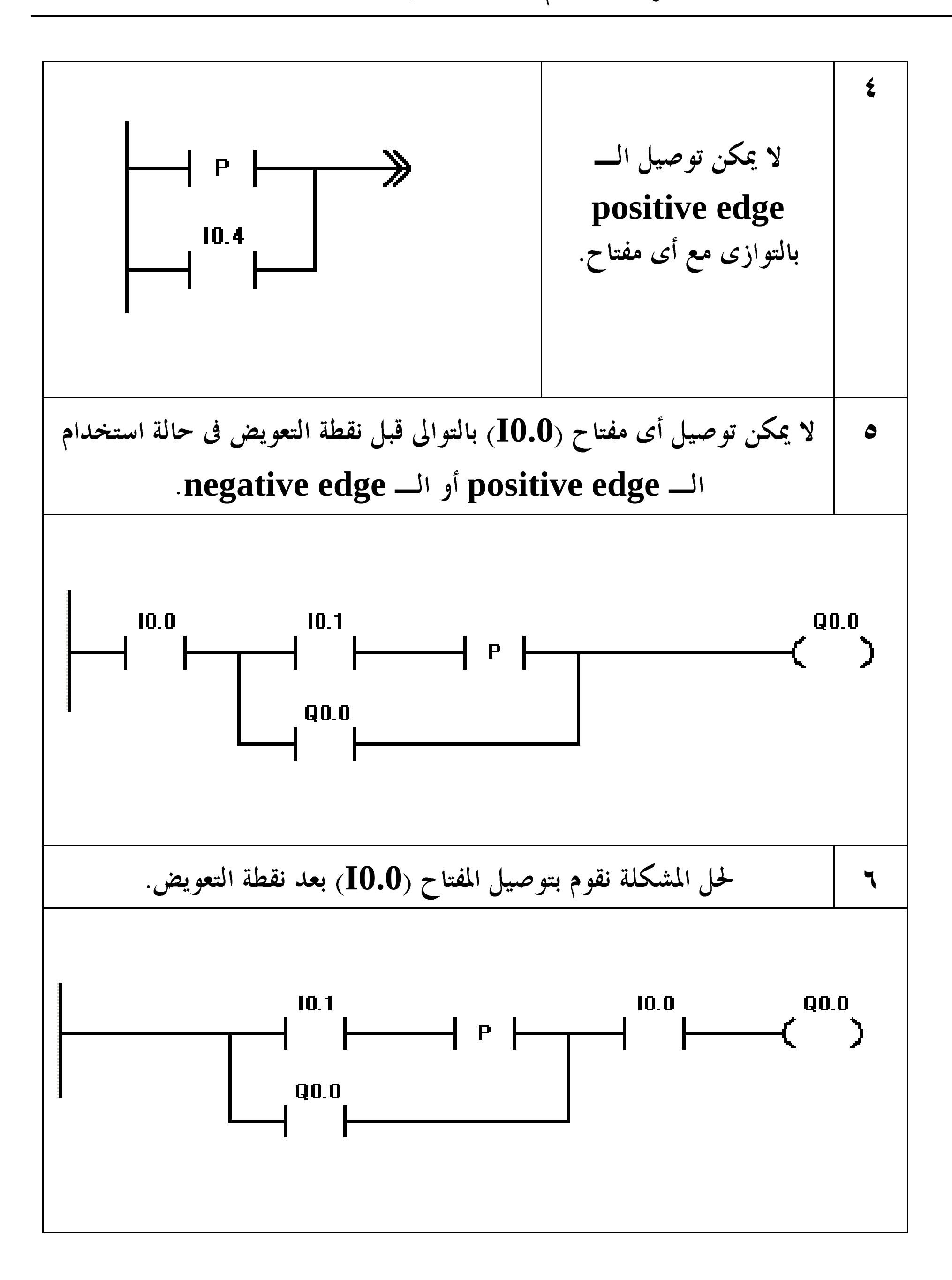
رسم تخطيطى:



ملاحظة:

توجد بعض التوصيلات الخاصة بالـــ positive edge و الـــ negative edge التى لا يجب تصميمها وسوف يتم توضيحها فى الجدول التالى لتجنب الأخطاء.

م	الشرح	الرسم
١	**لا يمكن توصيل الـــ positive edge دون توصيل مفتاح أولاً.**	P
٢	**لا يمكن توصيل الـــ negative edge دون توصيل مفتاح أولاً.**	N
٣	**لا يمكن توصيل الـــ negative edge بالتوازى مع أى مفتاح.**	N Q0.6 /



٤	لا يمكن توصيل الـ **positive edge** بالتوازى مع أى مفتاح.	
٥	لا يمكن توصيل أى مفتاح (**I0.0**) بالتوالى قبل نقطة التعويض فى حالة استخدام الـ **positive edge** أو الـ **negative edge**.	
٦	لحل المشكلة نقوم بتوصيل المفتاح (**I0.0**) بعد نقطة التعويض.	

الـــ Set/Reset .

رقم	الأسم	الرمز
١	Set	??.? (S) ????
٢	Reset	??.? (R) ????

الـــ Set:

??.?
(S)
????

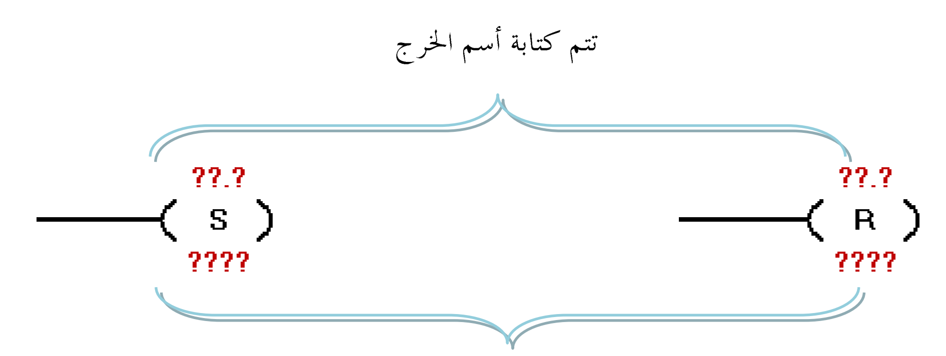
وهو يستخدم فى التشغيل, أي فى حالة إرسال إشارة إلى الـــ Set الخاص بأى output فأنة يعمل.

الـــ Reset:

??.?
(R)
????

وهو يستخدم فى الفصل, أي فى حالة إرسال إشارة إلى الـــ Reset الخاص بأى output فأنه يفصل.

ملاحظة:



تتم كتابة عدد المخرجات المراد التعامل معها بداية من الخرج المكتوب فوق , وكأنهم متصلون بالتوازى

فمثلاً فى المثال التالى:

- بالضغط على I0.1 سوف يعمل الخرج Q0.0 و Q0.1 و Q0.2 معاً.

I0.1 Q0.0
S
3

- بالضغط على I0.0 سوف يفصل الخرج Q0.0 و Q0.1 و Q0.2 معاً.

I0.0 Q0.0
R
3

مقارنة بين الـــ output العادى و الـــ output الـــ set/reset.

عدد النقاط	الإختلاف	الـــ output العادى	الـــ output الـــ set/reset.
١	الشكل	??.? ( )	??.? (S) ???? ??.? (R) ????
٢	نقطة الحفظ	يمكن استخدامها.	لا يحتاج إلى نقطة الحفظ لأنة يحتفظ بحالة.
٣	التكرار	لا يمكن تكرار الـــ output أكثر من مرة.	يجب تكرار الـــ output مرة مع الـــ set وأخرى مع الـــ reset.
٤	عدد المخرجات	لا يمكن تشغيل أكثر من output ألا عن طريق التوصيل بالتوازى.	يمكن تشغيل أكثر من output عن طريق كتابة العدد تحت الـــ Set أو الـــ Reset.
٥	المفاتيح	فى حالة الفصل يستخدم مفتاح مغلق.	فى حالة الفصل يستخدم مفتاح مفتوح.
٦	الأولوية	لا توجد أولوية لأن الـــ output لا يتكرر.	توجد أولوية لأن الـــ output يتكرر.

مثال:

✓ محرك يعمل من مكان واحد و يقف من مكان واحد.

عدد الدخل	نوع الدخل	أسم الدخل
١	n.o.	I0.1/S1
٢	n.o.	I0.2/S2
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتاكتور	Q0.0/K1M

شرح التمرين:

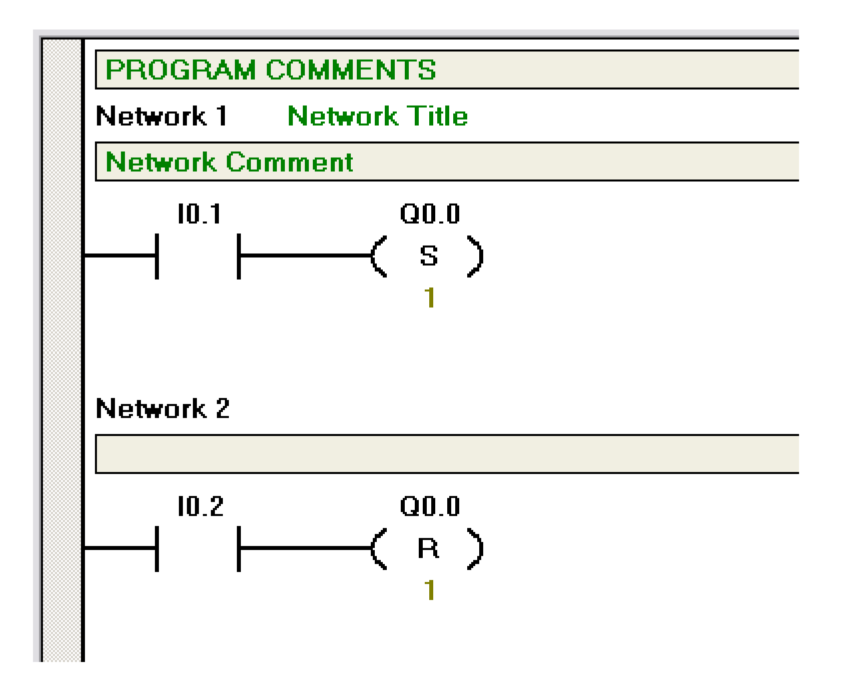
Network1

فى حالة الضغط على I0.1 تمر الإشارة إلى الـــ Set فتعمل Q0.0 فقط بحيث أن رقم واحد المكتوب أسفل الـــ Set يعنى أن Q0.0 سوف تعمل "لوحدها".

Network2

فى حالة الضغط على I0.2 تمر الإشارة إلى الـــ Reset فتفصل Q0.0 فقط بحيث أن رقم واحد المكتوب أسفل الـــ Reset يعنى أن Q0.0 سوف تفصل "لوحدها".

البرنامج:


PROGRAM COMMENTS
Network 1
Network Title
Network Comment
I0.1
Q0.0
S
1
Network 2
I0.2
Q0.0
R
1

مثال أخر:

- ✓ محركين يعملان بنفس الشروط, بحيث بالضغط على أحدى المفتاحين يعمل المحركين معاً و بالضغط على أحدى المفتاحين الأخريين يقف المحركين معاً.

المدخلات و المخرجات المستخدمة:

عدد الدخل	نوع الدخل	أسم الدخل
١	n.o.	I0.1/S1
٢	n.o.	I0.2/S2
٣	n.o.	I0.3/S3
٤	n.o.	I0.4/S4
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتاكتور	Q0.1/K1M
٢	كونتاكتور	Q0.2/K2M

شرح التمرين:

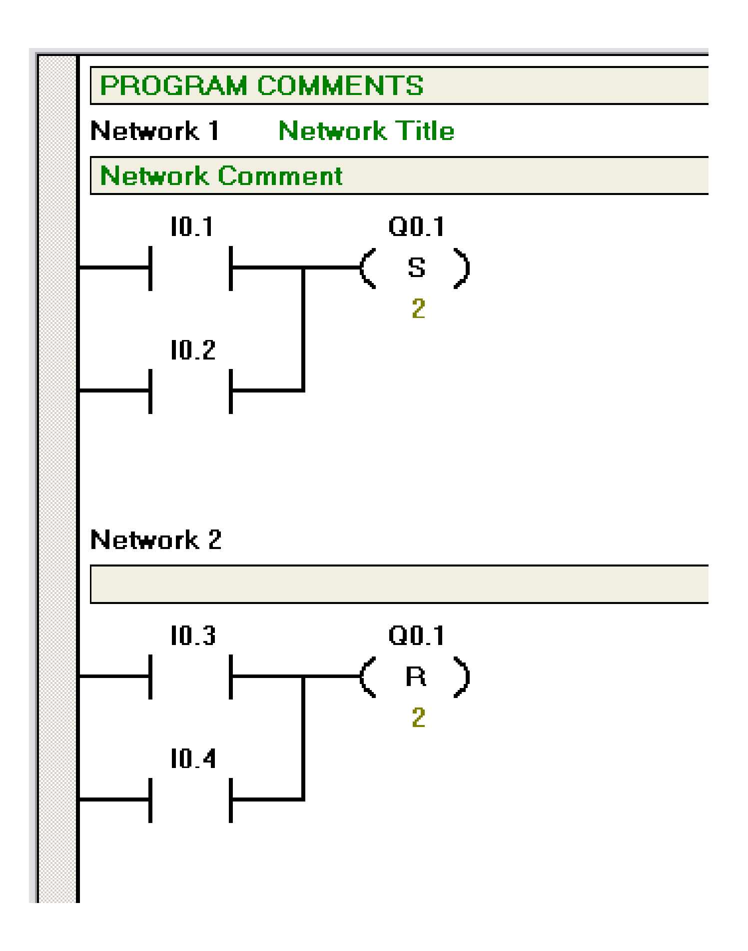
Network1

فى حالة الضغط على I0.1 أو I0.2 تمر الإشارة إلى الــ Set فتعمل Q0.1 و Q0.2 فقط بحيث أن رقم أثنين المكتوب أسفل الــ Set يعنى أن Q0.1 سوف تعمل ومعها الخرج التالى لها.

Network2

فى حالة الضغط على I0.3 و I0.4 تمر الإشارة إلى الــ Reset فتفصل Q0.1 و Q0.2 فقط بحيث أن رقم أثنين المكتوب أسفل الــ Reset يعنى أن Q0.1 سوف تفصل هى و الخرج التالى لها.

البرنامج:



ملاحظة:

تم توصيل المفتاحين المستخدمين في الفصل (Reset) على التوازى لأن الــ Reset يعمل عندما يستقبل الإشارة من أحدى أو كلتا المفتاحين.

مثال أخر باستخدام الــ Set/Reset:

✓ محرك يعمل فى اتجاهين باستخدام مفتاحين للتحكم باتجاه الحركة و مفتاح واحد للإيقاف.

عدد الدخل	نوع الدخل	أسم الدخل
١	n.o.	I0.1/S1
٢	n.o.	I0.2/S2
٣	n.o.	I0.3/S3
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتاكتور	Q0.1/K1M
٢	كونتاكتور	Q0.2/K2M

شرح التمرين:

Network1

فى حالة الضغط على I0.1 تمر الإشارة إلى الــ Set فتعمل Q0.1 فقط لأن رقم واحد المكتوب أسفل الــ Set يعنى أن Q0.1 سوف تعمل وحدها. ولكن لا يمكن أن يعمل الخرج Q0.1 أثناء عمل Q0.2

Network2

فى حالة الضغط على I0.3 تمر الإشارة إلى الــ Reset فتقف Q0.1 فقط لأن رقم واحد المكتوب أسفل الــ Reset يعنى أنQ0.1 سوف تقف وحدها.

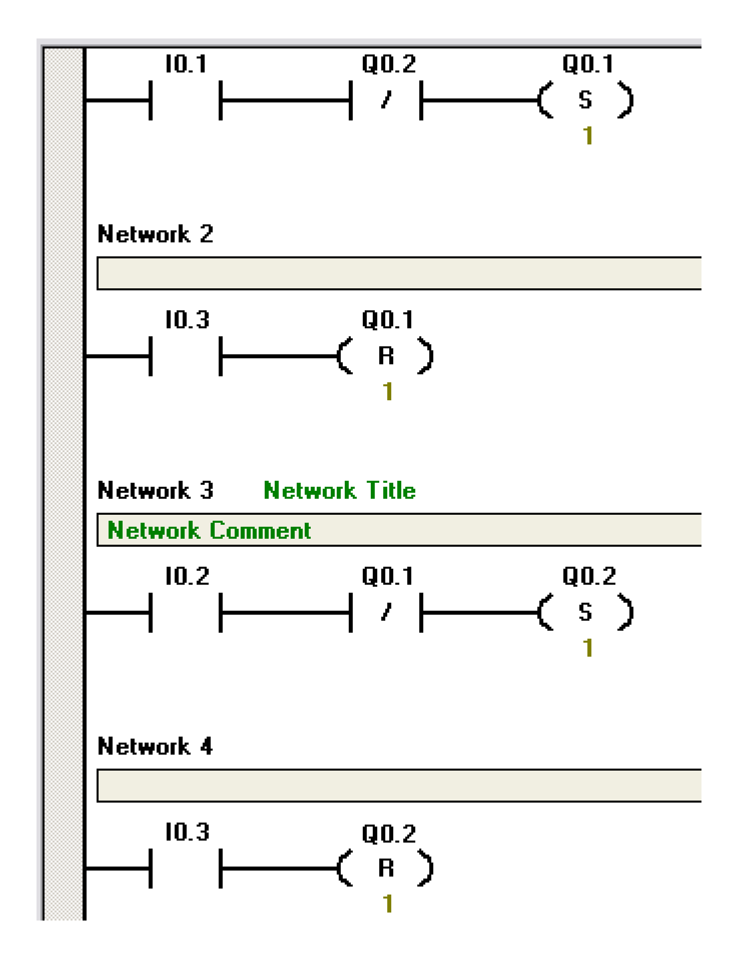
Network3

في حالة الضغط على I0.2 تمر الإشارة إلى الـــ Set فتعمل Q0.2 فقط لأن رقم واحد المكتوب أسفل الـــ Set يعنى أن Q0.2 سوف تعمل وحدها. ولكن لا يمكن أن يعمل الخرج Q0.2 أثناء عمل Q0.1

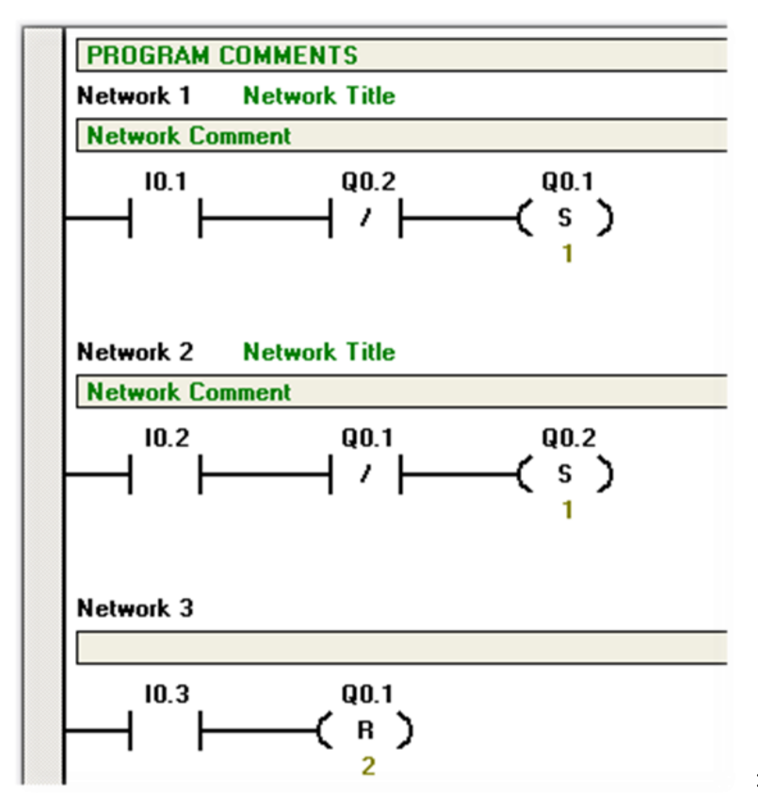
Network4

في حالة الضغط على I0.3 تمر الإشارة إلى الـــ Reset فتقف Q0.2 فقط لأن رقم واحد المكتوب أسفل الـــ Reset يعنى أن Q0.2 سوف تقف وحدها.

البرنامج:



طريقة أخرى



شرح التمرين:

Network1:

في حالة الضغط على I0.1 تمر الإشارة إلى الــ Set فتعمل Q0.1 فقط لأن رقم واحد المكتوب أسفل الــ Set يعني أن Q0.1 سوف تعمل وحدها. ولكن لا يمكن أن يعمل الخرج Q0.1 أثناء عمل Q0.2

Network2:

في حالة الضغط على I0.2 تمر الإشارة إلى الــ Set فتعمل Q0.2 فقط لأن رقم واحد المكتوب أسفل الــ Set يعني أن Q0.2 سوف تعمل وحدها. ولكن لا يمكن أن يعمل الخرج Q0.2 أثناء عمل Q0.1

Network3:

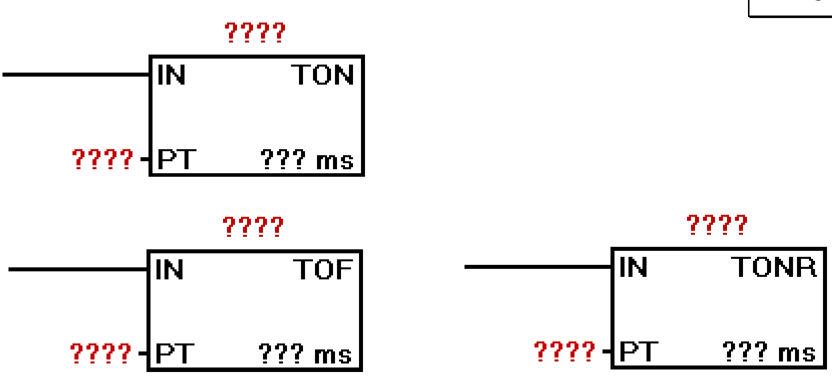
في حالة الضغط على I0.3 تمر الإشارة إلى الــ Reset الخاص بــ Q0.1 و Q0.2 لأن رقم أثنين المكتوب أسفل الــ Reset يعني أن سواء كان يعمل المحرك يميناً أو شمالاً سوف يقف.

الباب الخامس

المؤقتات الزمنية

- أنـــــــــــــــــــواع المؤقتات داخل الــ **PLC**.
- مؤقـــــــــــت التشغيل المتأخـــــــــ**TON**ــــــــــــر.
- مؤقـــــــــــت الفصل المتأخـــــــــ**TOF**ــــــــــــــر.
- مؤقـــــت التشغيل المتـــأخـــر المتـــ**TONR**ـــــد.
- كـــيـــفـــيـــة تغير دقة المؤقت الزمنـــــــــــــــــــى.
- مسميـــــــــات المؤقتات الزمنية فى الــ **CPU 224**.
- مسميـــــــــات المؤقتات الزمنية فى الــ **CPU 214**.
- رســـــــــــــــــم تخطيطى للمؤقتات الزمنية الثلاثة.
- تماريـــن تطبيقية على المؤقتـــــــــات الزمنيـــــــة.

المؤقتات الزمنية:



المؤقتات الزمنية الأكثر استخداماً فى جهاز الــ **PLC** هى:

١- مؤقت التشغيل المتأخر (Timer On Delay)TON.

٢- مؤقت الفصل المتأخر (Timer Off Delay)TOF.

٣- مؤقت التشغيل المتأخر الممتد (Retentive Timer On Delay)TONR.

المسميات الخاصة بالمؤقتات الزمنية الــ TON/TOF:

م	النوع	الوقت	الدقة	أقصى زمن	أسم المؤقتات
١	TON/TOF	١ثانية = ١٠٠٠	١ms	٣٢ ,٧٦٧ sec.	T32 ; T96
٢	TON/TOF	١ثانية = ١٠٠	١٠ms	٣٢٧ ,٦٧ sec.	T33 → T36 ; T97 → T100
٣	TON/TOF	١ثانية = ١٠	١٠٠ms	٣٢٧٦ ,٧ sec.	T37 → T63 ; T101 → T255

المسميات الخاصة بالمؤقتات الزمنية الـــ TONR:

م	النوع	الوقت	الدقة	أقصى زمن	أسم المؤقتات
١	TONR	١ثانية = ١٠٠٠	ms١	٣٢ ,٧٦٧	T0 ; T64
٢	TONR	١ثانية = ١٠٠	ms١٠	٣٢٧ ,٦٧	T1 → T4 ; T65 → T68
٣	TONR	١ثانية = ١٠	ms١٠٠	٣٢٧٦ ,٧	T5 → T31 ; T69 → T95

خصائص المؤقتات الزمنية:

أولاً:Integer ← أى أن الأرقام المستخدمة مع المؤقتات الزمنية يجب أن تكون أرقام صحيحة فقط وليست أرقام عشرية.

ثانياً:Word ← أى أن الأرقام المستخدمة مع المؤقتات الزمنية تكتب على ذاكرة مكونة من 16bits وليس أكثر ولا أقل.

ثالثاً:Signed ← أى أن الأرقام المستخدمة مع المؤقتات الزمنية يمكن أن تكون أرقام موجبة أو أرقام سالبة مع العلم أنه لا يوجد زمن بالسلب ولكن سنوضح فيما بعد السبب.

ملاحظة:

١- أقصى زمن للمؤقت الزمنى هو أقصى رقم موجب يمكن أن يكتب على ذاكرة "word" وهو من +٣٢٧٦٧ إلى 32768-.

٢ - لا يوجد مؤقت زمنى TOF فى الـــ CPU214.

٣- لاختيار الدقة المطلوبة يكفى كتابة أسم المؤقت الزمنى حسب الجدول السابق.

توضيح لمعرفة دقة المؤقت الزمني الخاص بالثلاث مؤقتات الزمنية.

م	توضيح	الشكل
١	مؤقت زمني TON أسمة T32 بزمن مسبق يساوى ٥ ثواني و بدقة ١ms.	T32 IN TON 5000 PT 1 ms
٢	مؤقت زمني TON أسمة T102 بزمن مسبق يساوى ٥ ثواني و بدقة ١٠٠ms.	T102 IN TON 50 PT 100 ms
٣	مؤقت زمني TOF أسمة T97 بزمن مسبق يساوى ٢ ثانية و بدقة ١٠ ms.	T97 IN TOF 200 PT 10 ms
٤	مؤقت زمني TOF أسمة T255 بزمن مسبق يساوى ٢ ثانية و بدقة ١٠٠ ms.	T255 IN TOF 20 PT 100 ms
٥	مؤقت زمني TONR أسمة T0 بزمن مسبق يساوى ١ ثانية و بدقة ١ms.	T0 IN TONR 1000 PT 1 ms
٦	مؤقت زمني TONR أسمة T5 بزمن مسبق يساوى ١ ثانية و بدقة ١٠٠ms.	T5 IN TONR 10 PT 100 ms

شرح كيفية عمل المؤقتات:

م	الأسم	الشرح	الشكل
١	TON مؤقت التشغيل المتأخر	يقوم بتغير وضعية الأقطاب الخاصة به بعد زمن من التشغيل أى انه يعتمد على الوقت المسبق. فى حاله انقطاع تغذية المؤقت الزمنى فأنه يعود إلى الصفر وبالضغط مرة أخرى يبدأ من البداية. يمكن تكرار نقاط المؤقت الزمنى دون حدود.	IN TON PT ??? ms ???? ????
٢	TOF مؤقت الفصل المتأخر	يقوم بتغير وضعية الأقطاب الخاصة به عند تشغيله و بعد زمن من قطع الإشارة يبدأ المؤقت الزمنى بالعمل إلى أن يصل إلى صفر فترجع وضعية الأقطاب كما كانت فى البداية قبل أن يعمل المؤقت الزمنى. الوقت بين فصل المؤقت الزمنى و	IN TOF PT ??? ms ???? ????

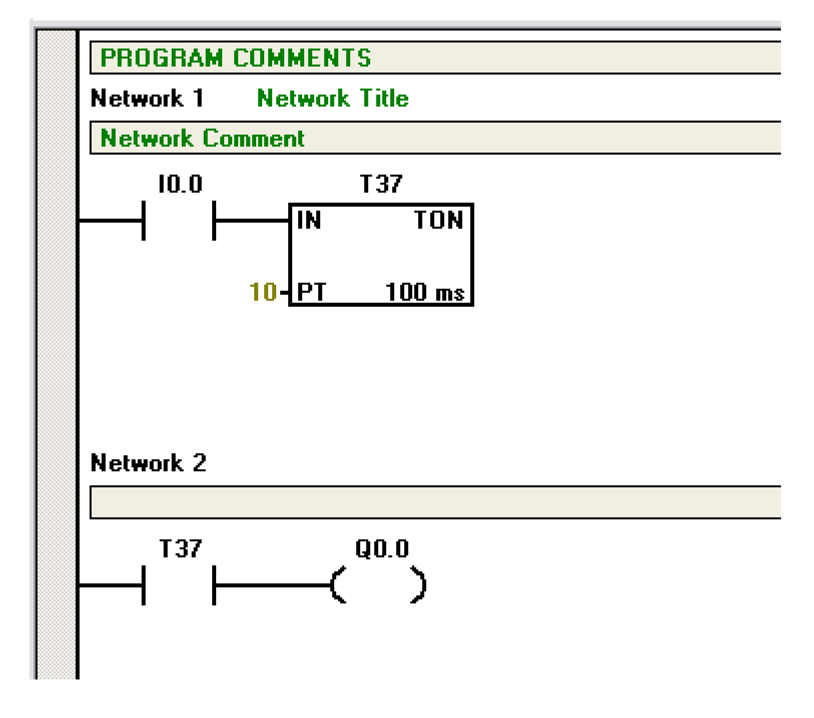
		رجوع وضعية الأقطاب إلى أصلها يعتمد على الوقت المسبق. يمكن تكرار نقاط المؤقت الزمنى دون حدود.	
٣	TONR مؤقت التشغيل المتأخر الممتد	يقوم بتغير وضعية الأقطاب الخاصة به بعد زمن من التشغيل أى انه يعتمد على الوقت المسبق. فى حاله انقطاع تغذية المؤقت الزمنى فأنه لا يعود إلى صفر وبالضغط مرة أخرى يستكمل العمل من نفس النقطة. يمكن تكرار نقاط المؤقت الزمنى دون حدود. لفصل المؤقت الزمنى نقوم بإرسال أشارة إلى الـــ reset الخاص بالمؤقت الزمنى.	???? IN TONR ???? PT ??? ms ??.? (R) ????

مثال TON :

محرك يعمل بعد زمن من الضغط على المفتاح.

عدد الدخل	نوع الدخل	أسم الدخل
١	n.o.	I0.0/S1
عدد المؤقتات الزمنية	**نوع المؤقتات الزمنية**	**أسم المؤقتات الزمنية**
١	TON	T37
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتكتور	Q0.0/K1M

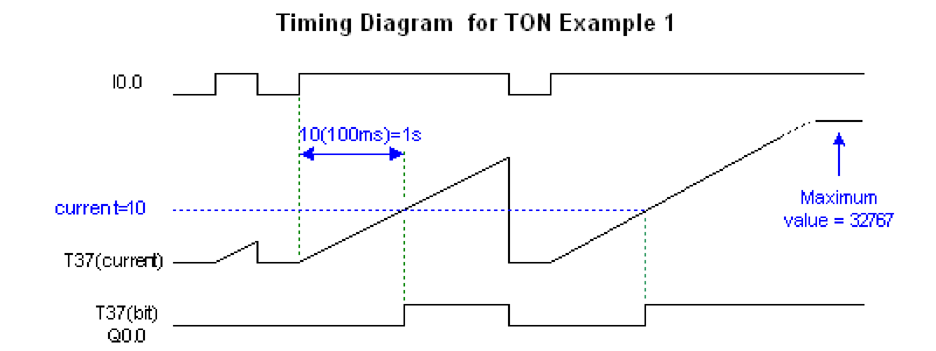
البرنامج:



الشرح:

بالضغط على المفتاح I0.0 يبدأ المؤقت الزمنى T37 فى العمل. عندما يصل إلى ثانية واحدة يعمل الخرج Q0.0 إلى أن يتم فصل المؤقت الزمنى T37 عن طريق فتح المفتاح I0.0

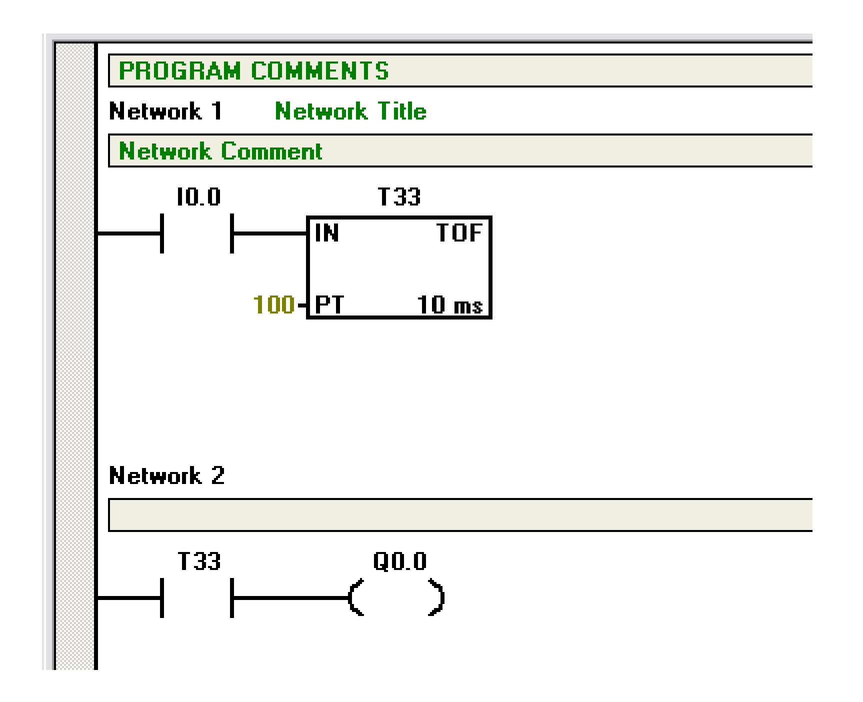
رسم تخطيطى:



مثال TOF :

محرك يعمل عند الضغط على المفتاح ولكن يفصل بعد زمن من فتح المفتاح.

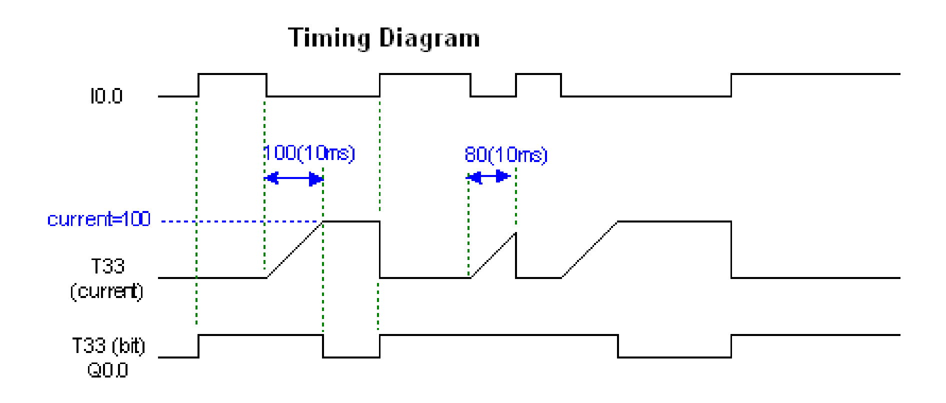
عدد الدخل	نوع الدخل	أسم الدخل
١	n.o.	I0.0/S1
عدد المؤقتات الزمنية	**نوع المؤقتات الزمنية**	**أسم المؤقتات الزمنية**
١	TOF	T33
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتكتور	Q0.0/K1M



الشرح:

بالضغط على المفتاح I0.0 يعمل المحرك ولكن عندما يفصل المفتاح يبدأ المؤقت الزمني T33 في العمل تنازلياً, بداياتاً من ثانية واحدة وحتى يصل المؤقت الزمني T33 إلى صفر يفصل الخرج Q0.0

رسم تخطيطي:



مثال TONR :

محرك يعمل بعد زمن من الضغط على المفتاح ولكن لا يفصل عند توقف المؤقت الزمني بل عندما ترسل أشارة reset إلى الـــ Timer.

عدد الدخل	**نوع الدخل**	**أسم الدخل**
١	n.o.	I0.0/S1
عدد المؤقتات الزمنية	**نوع المؤقتات الزمنية**	**أسم المؤقتات الزمنية**
١	TONR	T37
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتكتور	Q0.0/K1M

الشرح:

بالضغط على المفتاح I0.0 يبدأ المؤقت الزمنى T1 فى العمل. عندما يصل إلى ثانية واحدة يعمل الخرج Q0.0 إلى أن يتم فصل المؤقت الزمنى T1 عن طريق إرسال أشارة إلى الـــ reset الخاص بالمؤقت الزمنى بواسطة المفتاح I0.1

البرنامج:

PROGRAM COMMENTS

Network 1 Network Title

Network Comment

I0.0 — T1 (IN TONR, 100 – PT, 10 ms)

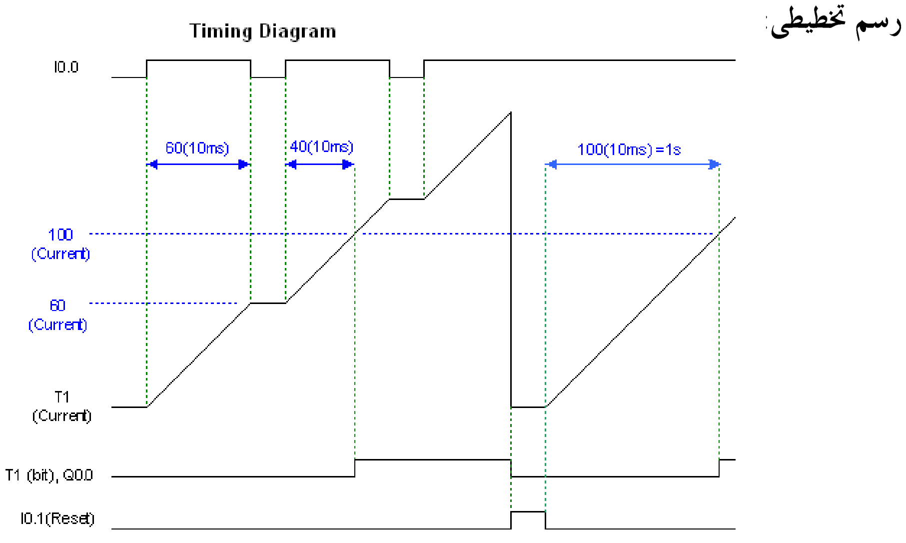
Network 2

T1 — Q0.0 ( )

Network 3

I0.1 — T1 (R) 1

رسم تخطيطي:



تمارين عملية على أنواع المؤقتات الزمنية..... Timers :

TON

١– قم بتنفيذ دائرة تحكم منطقية لمحركين بحيث أن المحرك الثانى يعمل أوتوماتيكياً بعد مرور زمن من تشغيل المحرك الأول يدوياً.

TOF

٢– قم بتنفيذ دائرة تحكم منطقية لمحركين بحيث أن المحرك الثانى يعمل أوتوماتيكياً بعد مرور زمن من إيقاف المحرك الأول يدوياً.

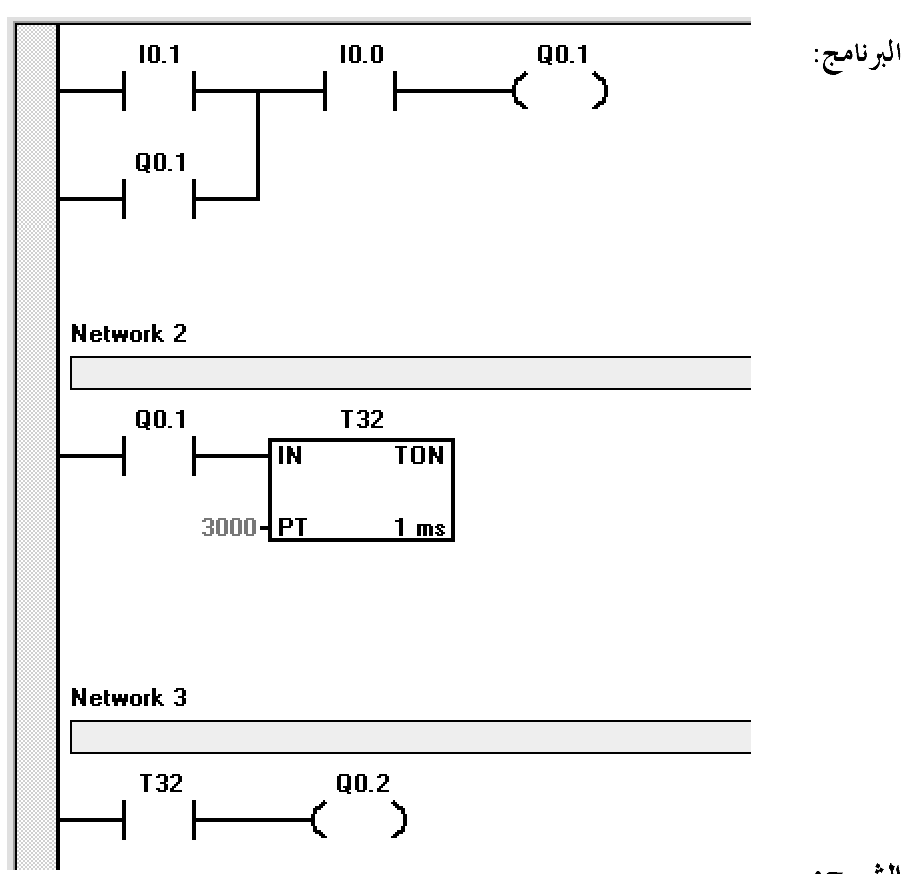
TONR

٣– بتنفيذ دائرة تحكم منطقية لمحركين بحيث أن المحرك الثانى يعمل أوتوماتيكياً بعد مرور زمن من تشغيل المحرك الأول يدوياً مع مراعاة أنه فى حالة فصل الأول وتشغيله مرة أخرى يستكمل العد على الزمن القديم و بعد أن يعمل المحرك الثانى يتم فصل المؤقت الزمنى يدوياً للاستخدام مرة أخرى فيما بعد.

✓ **التمرين الأول باستخدام مؤقت زمنى TON:**

عدد الدخل	نوع الدخل	أسم الدخل
١	n.c.	I0.0/S1
٢	n.o.	I0.1/S2
عدد المؤقتات الزمنية	**نوع المؤقتات الزمنية**	**أسم المؤقتات الزمنية**
١	TON	T32
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتكتور	Q0.1/K1M
٢	كونتكتور	Q0.2/K2M

البرنامج:



الشرح:

Network1:

بالضغط على I0.1 و مع مراعاة أن المفتاح I0.0 مغلق بالخارج (أنظر صفحة **110**) فأن Q0.1 سوف تعمل فى الحال.

Network2:

فى نفس الدورة cycle التى ستعمل فيها Q0.1 سوف يعمل أيضاً المؤقت الزمنى T32.

:Network3

بعد مرور ٣ ثوانى وعند مرور الـــ CPU على الـــ Network3 سوف يعمل Q0.2

✓ **التمرين الثانى باستخدام مؤقت زمنى TOF**:

عدد الدخل	نوع الدخل	أسم الدخل
١	n.c.	I2.0/S1
٢	n.c.	I2.2/S2
٣	n.o.	I3.1/S3
عدد المؤقتات الزمنية	**نوع المؤقتات الزمنية**	**أسم المؤقتات الزمنية**
١	TOF	T100
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتكتور	Q0.5/K1M
٢	كونتكتور	Q4.4/K2M

الشرح:

:Network1

بالضغط على I3.1 و مع مراعاة أن المفتاح I2.0 مغلق بالخارج (أنظر صفحة **110**) فأن Q4.4 سوف تعمل فى الحال.

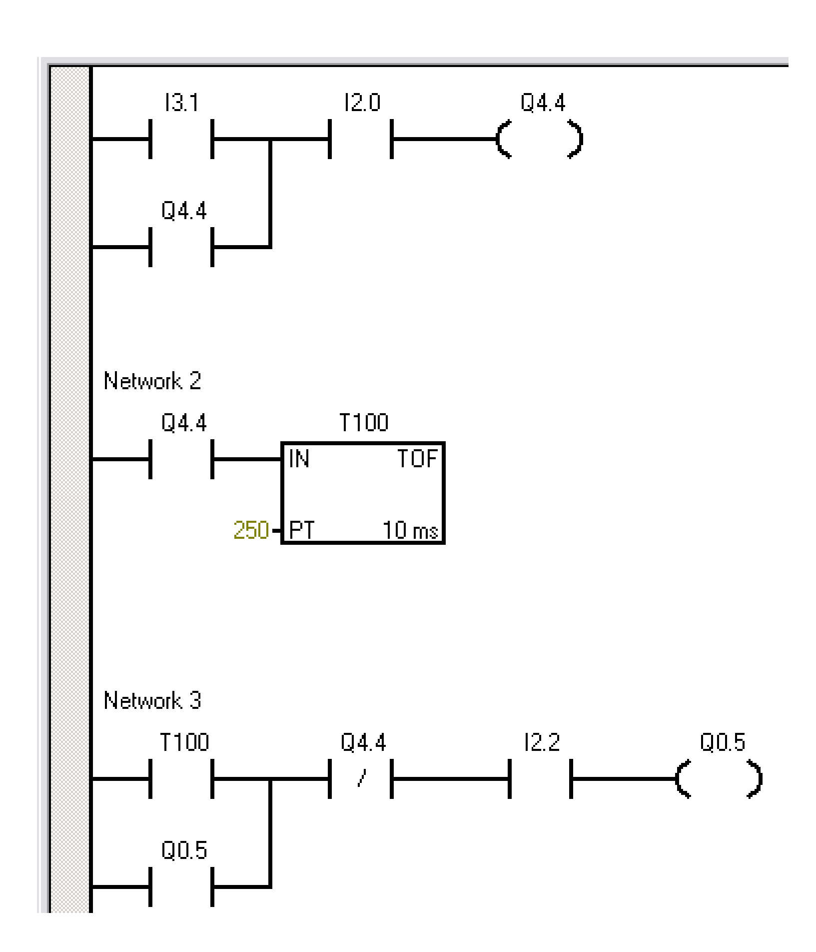
:Network2

فى نفس الدورة cycle التى ستعمل فيها Q4.4 سوف يعمل أيضاً المؤقت الزمنى T100 ويقوم بتغير وضعية الأقطاب الخاصة به ولكنه لن سيبدأ بالعد ألا عند انقطاع التغذية أى عند فصل Q4.4.

:Network3

عندما يتم فصل Q4.4 وبعد مرور ٥،٢ ثانية وعند مرور الــ CPU على الــ Network3 سوف يعمل Q0.2

البرنامج:


I3.1
I2.0
Q4.4
Q4.4
Network 2
Q4.4
T100
IN
TOF
250
PT
10 ms
Network 3
T100
Q4.4
I2.2
Q0.5
Q0.5

✓ **التمرين الثالث باستخدام مؤقت زمنى TONR**:

عدد الدخل	نوع الدخل	أسم الدخل
١	n.c.	I0.0/S1
٢	n.c.	I1.0/S2
٣	n.o.	I1.1/S3
عدد المؤقتات الزمنية	**نوع المؤقتات الزمنية**	**أسم المؤقتات الزمنية**
١	TONR	T0
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتكتور	Q2.2/K1M
٢	كونتكتور	Q2.4/K2M

الشرح:

:Network1

بالضغط على I1.1 و مع مراعاة أن المفتاح I1.0 مغلق بالخارج (أنظر صفحة **110**) فأن Q2.2 سوف تعمل فى الحال.

:Network2

فى نفس الدورة cycle التى ستعمل فيها Q2.2 سوف يعمل أيضاً المؤقت الزمنى T0.

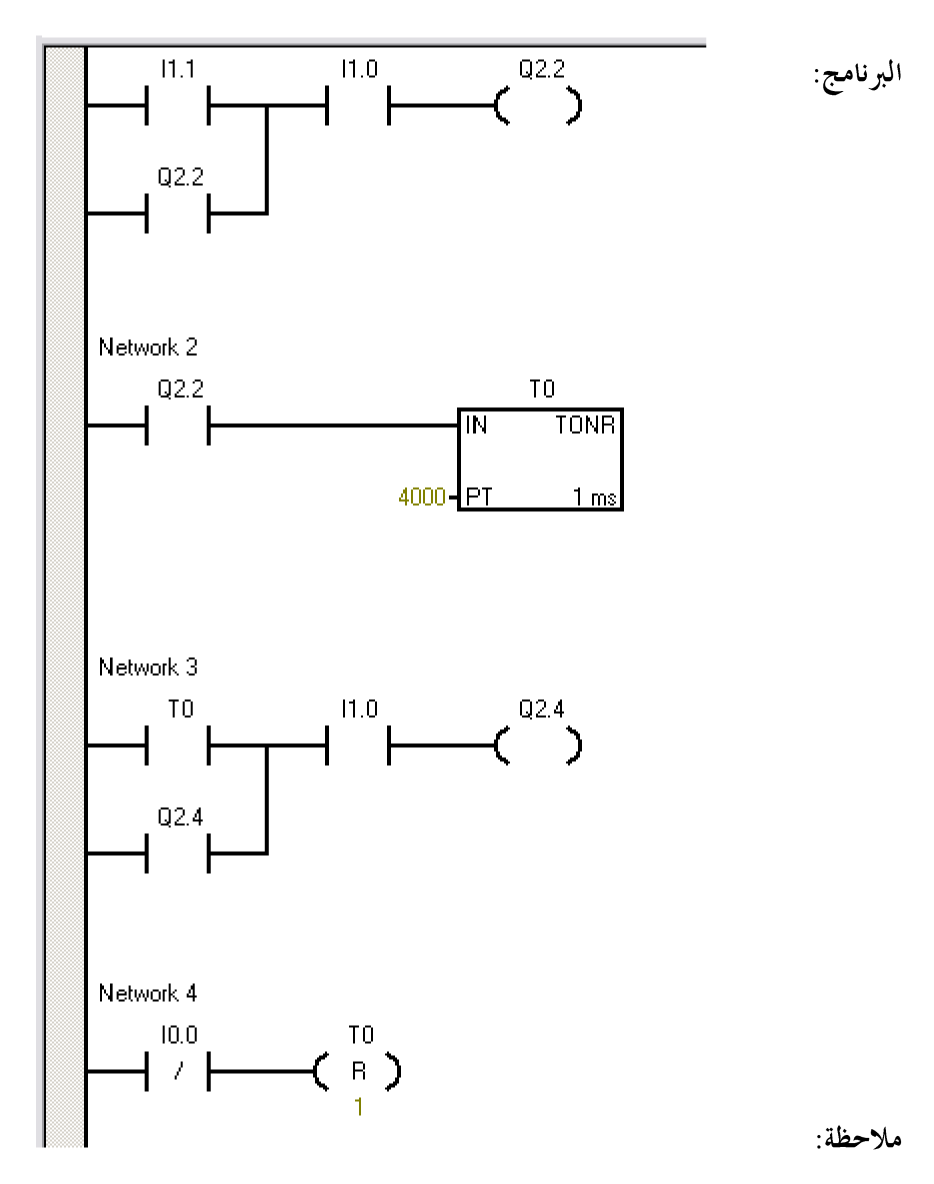
:Network3

بعد مرور ٤ ثوانى وعند مرور الــ CPU على الــ Network3 سوف يعمل Q2.4

:Network4

فى حاله فتح I0.0 بالخارج فأنه يغلق بالداخل فيعمل الــ Reset فيرجع المؤقت الزمنى إلى صفر ليبدأ من جديد.

البرنامج:



ملاحظة:

- يفضل وضع نقطة Positive edge على التوالى بعد I0.0 لكى لا تستمر أشارة الــ reset دائماً.
- في حاله فصل Q2.2 فأن المؤقت الزمني T0 لا يرجع إلى صفر بل يتوقف وعند أعادة تشغيل Q2.2 مرة أخرى فأن المؤقت الزمني T0 يستكمل من نفس القيمة التي قد توقف عندها قبل فصل Q2.2

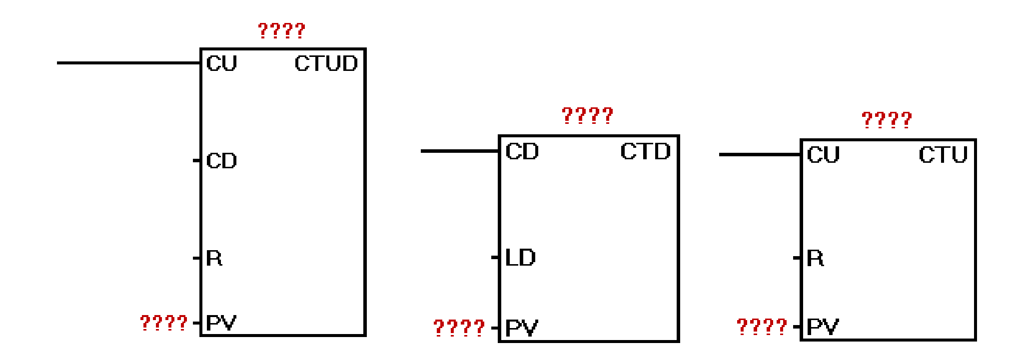
الباب السادس

العدادات

- أنــــــــــــواع العــدادات داخل الــ PLC.
- عـــــــــــــ**CTU**ــــــــــــــــــداد تصاعدى.
- عـــــــــــــ**CTD**ــــــــــــــــــداد تنازلـــى.
- عــــــ**CTUD**ـــــــــداد تصاعدى و تنازلـــى.
- خـــــصــــــــائص العــــــــــــــــــــداداتــ.
- مسميات العــــــــــــــــداداتــ فى الــ CPU 224.
- مسميات العــــــــــــــــداداتــ فى الــ CPU 214.
- رســــــــم تخطيطى للعــــــــــــــــــداداتــ الثلاثة.
- تماريــــــــــــــــن تطبيقية على العــــــــــداداتــ.

العدادات:

تستخدم العدادات بمختلف أنوعها فى كثير من التمرين العملية حيث يمكن العداد أن يستخدم لمعرفة عدد القطع التى تم تصنعها, لمعرفة عدد السيارات داخل الجراج, لتعبئة مجموعة زجاجات فى كرتونه و للكثير من العمليات المختلفة التى سوف نتعرض لجزء منها فى ما بعد.



العدادات الأكثر أستخداماً فى جهاز الـــ PLC هم:

١- عداد تصاعدى CTU (Counter Up).

٢- عداد تنازلى CTD (Counter Down).

٣- عداد تصاعدى و تنازلى CTUD (Counter Up and Down).

المسميات الخاصة بالعدادات الـــ CTU/CTD/CTUD(CPU 224):

م	النوع	العدات	الدقة	أقصى رقم موجب	أقصى رقم سالب	أسم العدادات
١	CTU	١عدة = ١	١:١	+٣٢٧٦٧	-٣٢٧٦٨	C0 → C255
٢	CTD	١ عدة = ١	١:١	+٣٢٧٦٧	-٣٢٧٦٨	C0 → C255
٣	CTUD	١ عدة = ١	١:١	+٣٢٧٦٧	-٣٢٧٦٨	C0 → C255

المسميات الخاصة بالعدادات الـــ CTU/CTD/CTUD(CPU 214):

م	النوع	العدات	الدقة	أقصى رقم موجب	أقصى رقم سالب	أسم العدادات
١	CTU	١عدة = ١	١:١	+٣٢٧٦٧	-٣٢٧٦٨	C0 → C47
٢	CTD	لا يوجد	لا	يوجـــــد	لا	يوجـــــد
٣	CTUD	١ عدة = ١	١:١	+٣٢٧٦٧	-٣٢٧٦٨	C48 → C127

خصائص العدادات الزمنية:

أولاً:**Integer** ← أى أن الأرقام المستخدمة مع العدادات يجب أن تكون أرقام صحيحة فقط.

ثانياً:**Word** ← أى أن الأرقام المستخدمة مع العدادات تكتب على ذاكرة مكونة من 16bits.

ثالثاً:**Signed** ← أى أن الأرقام المستخدمة مع العدادات يمكن أن تكون أرقام موجبة أو أرقام سالبة.

ملاحظة:

١ – أقصى رقم للعدادات هو أقصى رقم يمكن أن يكتب على ذاكرة word.

٢ – أمثلة عن شكل العدادات, فمثلاً:

م	توضيح	الشكل
١	عداد تصاعدى CTU أسمة C10 بقيمة مسبقة تساوى ٥ عدات.	C10 CU CTU R 5 PV
٢	عداد تنازلى CTD أسمة C29 بقيمة مسبقة تساوى ١٥ عدة.	C29 CD CTD LD 15 PV
٣	عداد تصاعدى و تنازلى CTUD أسمة C100 بقيمة مسبقة تساوى ٥٣ عدة.	C100 CU CTUD CD R 53 PV

شرح كيفية تشغيل العدادات:

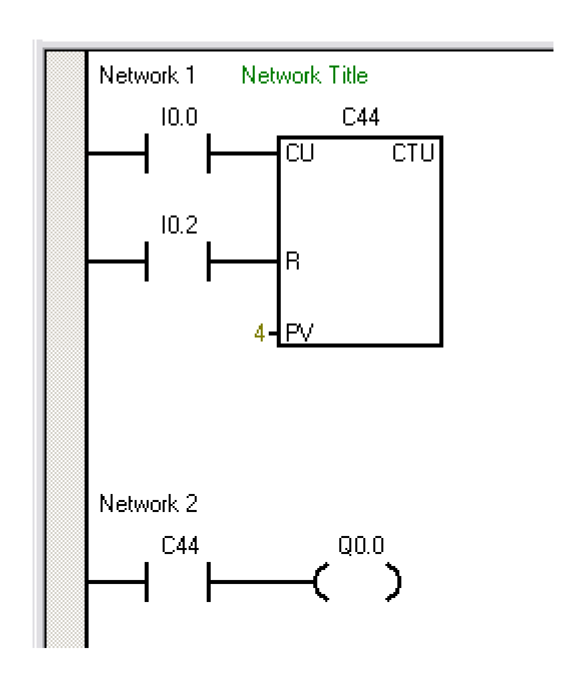
م	الأسم	الشرح	الشكل
١	**CTU** عداد تصاعدى	يقوم بتغير وضعية الأقطاب الخاصة به عندما يصل إلى القيمة المسبقة المحددة منذ البداية. فى حاله انقطاع تغذية العداد فأنع لا يعود إلى صفر بل بإرسال أشارة إلى الـــ R فأنة يعود إلى صفر ليبدأ من البداية. يمكن تكرار نقاط العداد دون حدود. العداد CTU يقوم بالعد تصاعدياً بداية من صفر إلى أن يصل أو يتخطى القيمة المسبقة.	???? CU CTU R ???? PV
٢		يقوم بتغير وضعية الأقطاب الخاصة به عندما يصل إلى	

	CTD عداد تنازلى	صفر. فى حاله انقطاع تغذية العداد فأنع لا يعود إلى القيمة المسبقة بل بإرسال إشارة إلى الــ LD فأنة يعود إلى القيمة المسبقة ليبدأ من البداية. يمكن تكرار نقاط العداد دون حدود. العداد CTD يقوم بالعد تنازلياً بداية من القيمة المسبقة إلى أن يصل إلى صفر.	???? CD CTD LD ???? PV
٣	CTUD عداد تصاعدى و تنازلى	يقوم بتغير وضعية الأقطاب الخاصة به عندما يصل إلى القيمة المسبق المحددة منذ البداية. فى حاله انقطاع تغذية العداد فأنع لا يعود إلى صفر بل بإرسال أشارة إلى الــ R فأنة يعود إلى صفر ليبدأ من البداية	???? CU CTUD CD R ???? PV

		يمكن تكرار نقاط العداد دون حدود. العداد CTUD يقوم بالعد تصاعدياً و تنازلياً بداية من صفر إلى أن يصل أو يتخطى القيمة المسبقة.	

مثال CTU:

✓ محرك يعمل بعد أربع عدات.

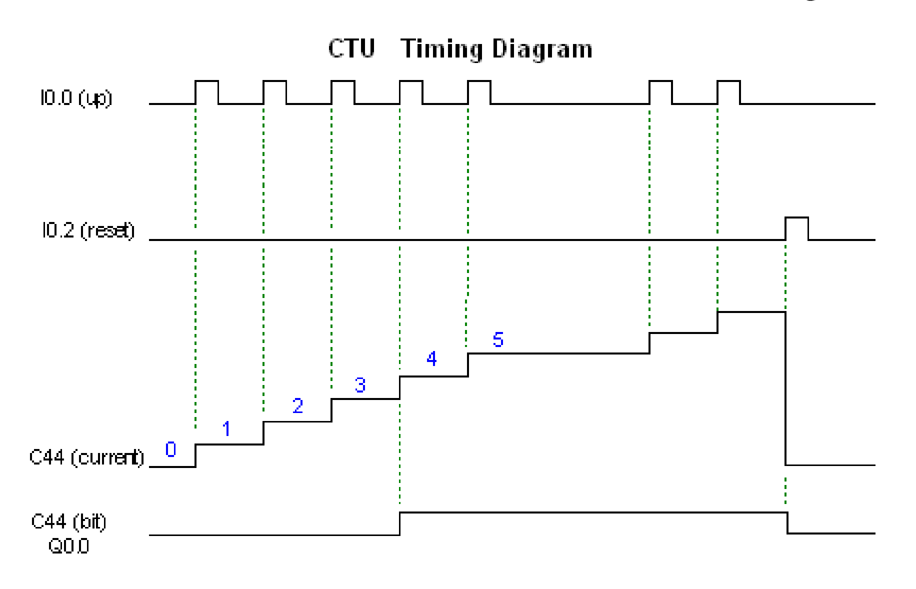


عدد الدخل	نوع الدخل	أسم الدخل
١	n.o.	I0.0/S1
٢	n.o.	I0.2/S1
عدد العدادات	**نوع العدادات**	**أسم العدادات**
١	CTU	C44
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتاكتور	Q0.0/K1M

الشرح:

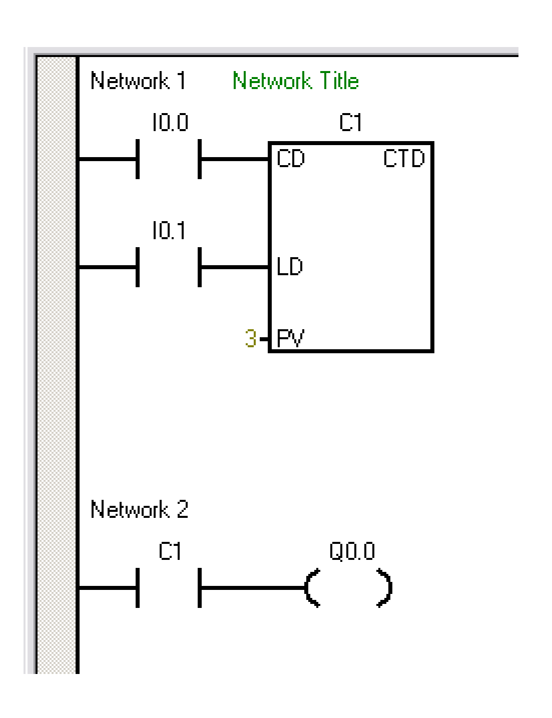
بالضغط على المفتاح I0.0 يبدأ العداد بالعد تصاعدياً وبعد أربع أشارات متفرقة يكون قد وصل العداد C44 إلى أربعة فيقوم بتغير النقاط المفتوحة فيعمل الخرج Q0.0 إلى ان يتم تصفير العداد بواسطة المفتاح I0.2 للبدء من جديد.

رسم تخطيطي:



مثال CTD:

✓ محرك يعمل بعد ثلاث عدات.

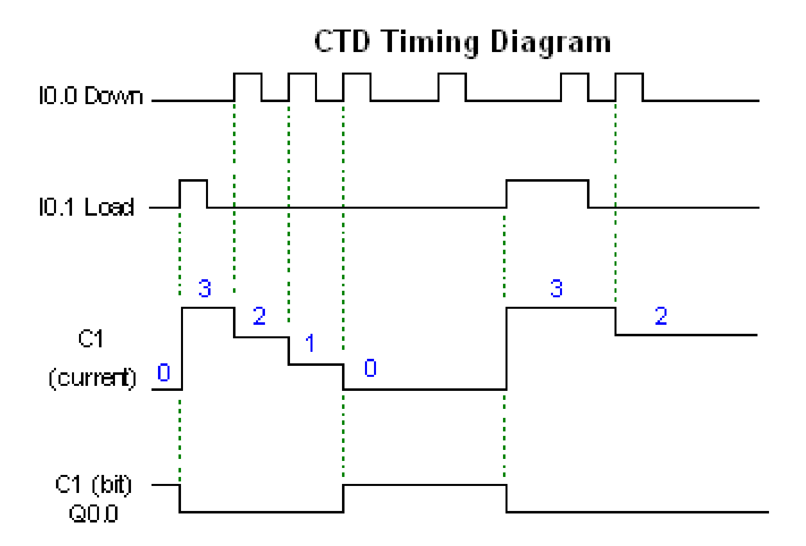


عدد الدخل	نوع الدخل	أسم الدخل
١	n.o.	I0.0/S1
٢	n.o.	I0.1/S1
عدد العدادات	**نوع العدادات**	**أسم العدادات**
١	CTD	C1
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتاكتور	Q0.0/K1M

الشرح:

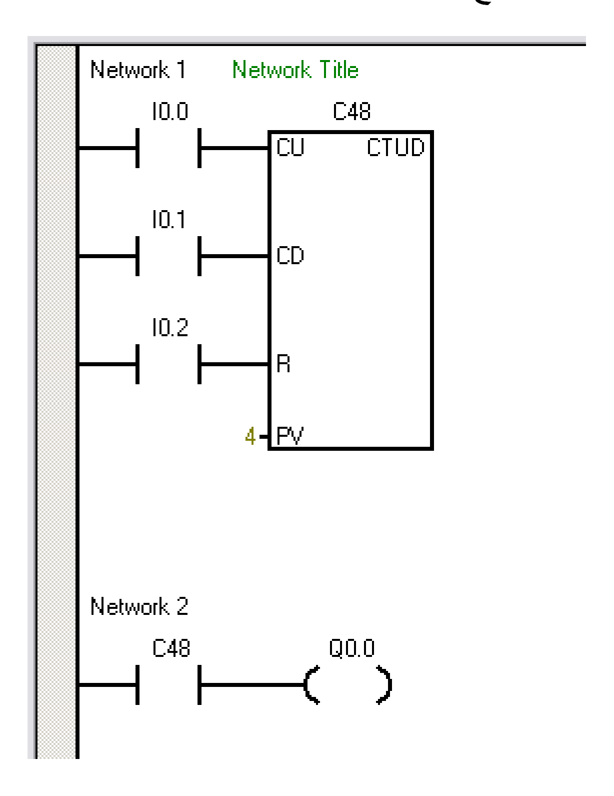
بالضغط على المفتاح I0.0 يبدأ العداد بالعد تنازلياً وبعد ثلاث أشارات متفرقة يكون قد وصل العداد C1 إلى صفر فيقوم بتغير النقاط المفتوحة فيعمل الخرج Q0.0 إلى ان يتم أعادة الرقم إلى ثلاثة بواسطة المفتاح I0.1 للبدء من جديد.

رسم تختيطى:



مثال CTUD :

✓ محرك يعمل بعد أربع عدات.

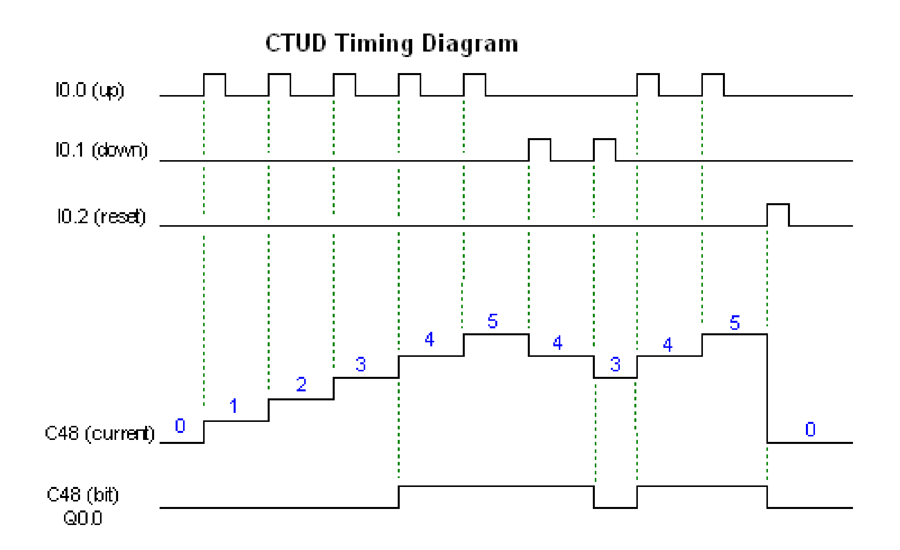


عدد الدخل	نوع الدخل	أسم الدخل
١	n.o.	I0.0/S1
٢	n.o.	I0.1/S1
عدد العدادات	نوع العدادات	أسم العدادات
١	CTUD	C48
عدد الخرج	نوع الخرج	أسم الخرج
١	كونتاكتور	Q0.0/K1M

الشرح:

بالضغط على المفتاح I0.0 يبدأ العداد بالعد تصاعدياً بينما بالضغط على المفتاح I0.1 يبدأ العداد بالعد تنازلياً وبعد وصول العداد إلى أربعة سواء بالضغط أربع مرات على I0.0 أربع مرات أو بأى طريقة أخرى بحيث يكون قد وصل العداد C44 إلى أربعة فيقوم بتغير النقاط المفتوحة فيعمل الخرج Q0.0 إلى ان يتم تصفير العداد بواسطة المفتاح I0.2 للبدء من جديد.

رسم تخطيطى:



تمارين عملية على أنواع العدادات (counters):

CTU

١- قم بتنفيذ دائرة تحكم منطقية لمحركين بحيث أن المحرك الثانى يعمل أوتوماتيكياً بعد أن يكون قد عمل المحرك الأول يدوياً خمس مرات متتالية بأستخدام عداد تصاعدى.

CTD

٢- قم بتنفيذ دائرة تحكم منطقية لمحركين بحيث أن المحرك الثانى يعمل أوتوماتيكياً بعد أن يكون قد عمل المحرك الأول يدوياً خمس مرات متتالية باستخدام عداد تنازلى.

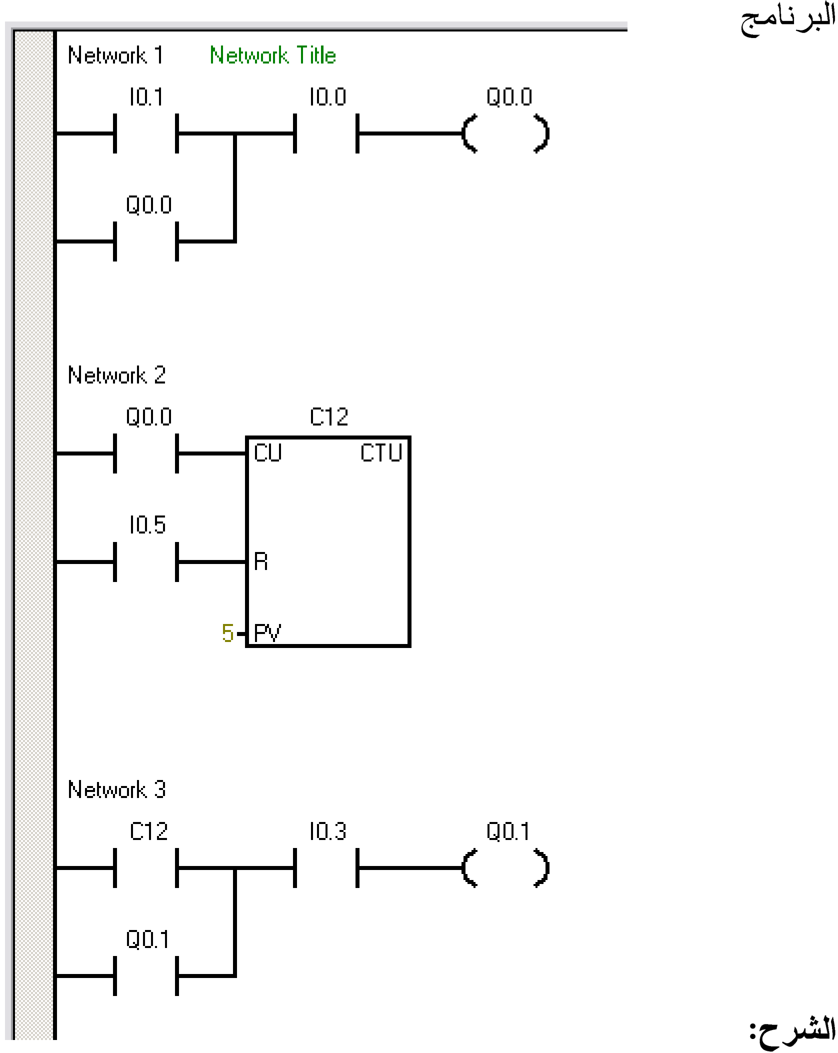
CTUD

٣- بتنفيز دائرة تحكم منطقية لخرجين (لمبات أشارة) بحيث أن الإشارة الخضراء تشير إلى أن الجراچ فارغ بينما تشير الإشارة الحمراء إلى أن الجراچ ممتلئ. علماً أن أقصى عدد سيارات داخل الجراچ هو خمسة فأنة عندما يصل العدد إلى خمسة يجب أن تضاء الإشارة الحمراء لتشير أن عدد السيارات داخل الجراچ أكتمل.

- ✓ **التمرين الأول باستخدام عداد تصاعدى CTU:**

عدد الدخل	نوع الدخل	أسم الدخل
١	n.c.	I0.0/S1
٢	n.o.	I0.1/S2
٣	n.c.	I0.3/S3
٤	n.o.	I0.5/S4
عدد العدادات	**نوع العدادات**	**أسم العدادات**
١	CTU	C12
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتاكتور	Q0.0/K1M
٢	كونتاكتور	Q0.1/K2M

البرنامج



الشرح:

Network1

بالضغط على I0.1 و مع مراعاة أن المفتاح I0.0 مغلق بالخارج (أنظر صفحة 110) فأن Q0.0 سوف تعمل فى الحال.

Network2

فى نفس الـــ cycle التى ستعمل فيها Q0.0 سوف يعمل العداد C12 ويبدأ بالعد تصاعدياً.

Network3

بعد تكرار هذه العملية خمس مرات وعند مرور الـــ CPU على الـــ Network3 سوف يعمل Q0.1

✓ **التمرين الثانى باستخدام عداد تصاعدى CTD**:

عدد الدخل	نوع الدخل	أسم الدخل
١	n.c.	I0.0/S1
٢	n.o.	I0.1/S2
٣	n.c.	I0.3/S3
٤	n.o.	I0.5/S4
عدد العدادات	**نوع العدادات**	**أسم العدادات**
١	CTD	C33
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتاكتور	Q0.0/K1M
٢	كونتاكتور	Q0.1/K2M

الشرح:

Network1

بالضغط على I0.1 و مع مراعاة أن المفتاح I0.0 مغلق بالخارج (أنظر صفحة 110) فأن Q0.0 سوف تعمل فى الحال.

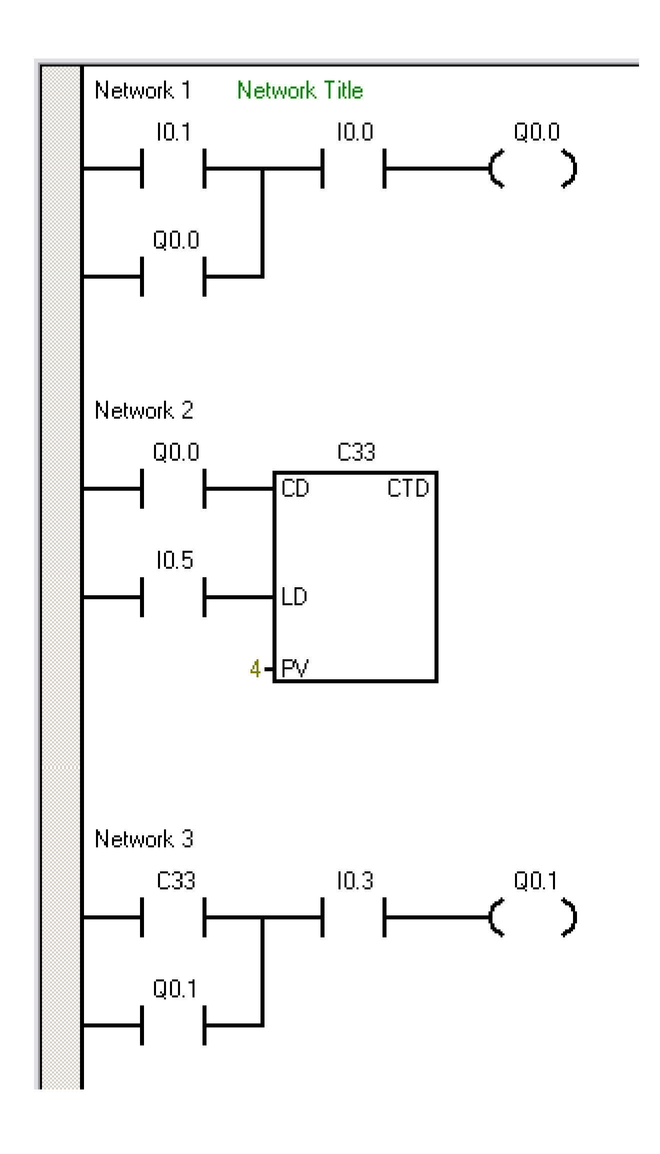
Network2

في نفس الـــ cycle التي ستعمل فيها Q0.0 سوف يعمل العداد C12 ويبدأ بالعد تنازلياً.

Network3

بعد تكرار هذه العملية أربع مرات وعند مرور الـــ CPU على الـــ Network3 سوف يعمل Q0.1

البرنامج



- ✓ **التمرين الثالث باستخدام عداد تصاعدى و تنازلى CTUD**:

عدد الدخل	نوع الدخل	أسم الدخل
١	n.c.	I0.0/S1
٢	n.o.	I0.1/S2
٣	n.c.	I0.3/S3
٤	n.o.	I0.5/S4
٥	n.o.	I0.6/S5
عدد العدادات	**نوع العدادات**	**أسم العدادات**
١	CTUD	C200
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	لمبة حمرة	Q0.1/H1
٢	لمبة خضرة	Q0.2/H2

الشرح:

Network1

بالضغط على I0.1 و مع مراعاة أن المفتاح I0.0 مفتوح بالخارج (أنظر صفحة 110) فأن العداد C200 سوف يبدأ بالعد تصاعدياً بينما بالضغط على I0.2 و مع مراعاة أن المفتاح I0.0 مفتوح بالخارج (أنظر صفحة 110) فأن العداد C200 سوف يبدأ بالعد تنازلياً.

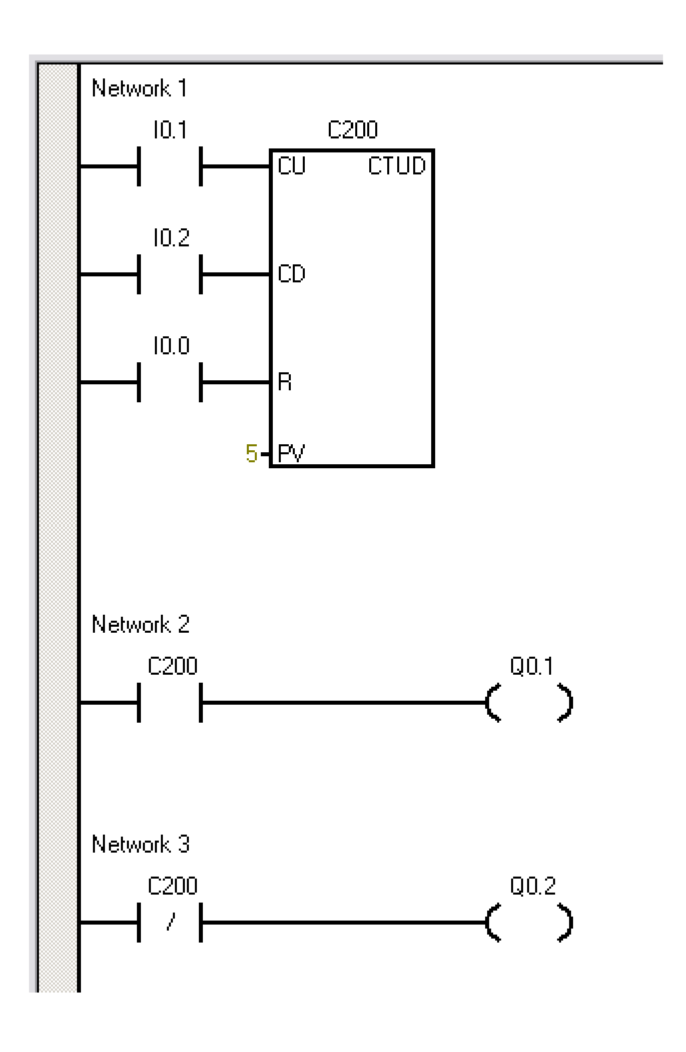
Network2

عندما يصل العداد إلى خمسة يقوم بغلق النقطة المفتوحة فتضئ الإشارة الحمراء لتشير أن الجراچ قد أمتلئ.

Network3

عندما يصل العداد إلى خمسة يقوم بفتح النقطة المغلقة فتضيء الإشارة الخضراء لتشير أن الجراچ لم يعد ممتلئ.

البرنامج



الباب السابع

المتغيرات

- أنـــــــــــــواع المتغيرات داخل الـــ PLC.
- متغيـــــــــــــــــــــــــرات بحجم bit.
- متغيـــــــــــــــــــــــــرات بحجم byte.
- متغيـــــــــــــــــــــــــرات بحجم word.
- متغيـــــــــــــــــــــــــرات بحجم Dword.
- طريقة أستخدام المتغيـــــــــــــــرات بحجم bit.
- مسميات المتغيـــــــــــــــرات فى الـــ PLC.
- تماريـــــــــــــن تطبيقية على المتغيـــــــــرات.

المتغيرات.........variables:

هى ذاكرة داخل الـــ PLC تستخدم فى كتابة أى أرقام حتى تستخدم فى ما بعد سواء فى عمليات حسابية أو فى المقارنة إلخ...

تنقسم الذاكرة المتغيرة variables إلى:

bit - byte - word - Dword

١- V(BIT): هى أصغر وحدة للذاكرة داخل جهاز الـــ PLC وهى قد تحتوى على صفر أو واحد.

1

٢- V(BYTE): هى ذاكرة داخل جهاز الـــ PLC وهى قد تحتوى على صفر و واحد وهى تتكون من 8 bits.

1	0	1	0	0	0	1	1

٣- V(WORD): هى ذاكرة داخل جهاز الـــ PLC وهى قد تحتوى على صفر و واحد وهى تتكون من 2 bytes و 16 bits.

0	1	1	1	0	1	0	0	1	0	0	1	0	1	1	1

٤- V(D.WORD): هى أكبر وحدة للذاكرة داخل جهاز الـــ PLC وهى قد يحتوى على صفر و واحد وهو تتكون من 2 words و 4 bytes و 32 bits.

1	1	1	0	0	1	0	1	1	1	0	1	0	1	1	1	0	1	1	1	0	1	0	0	1	0	0	1	0	1	1	1

هام:

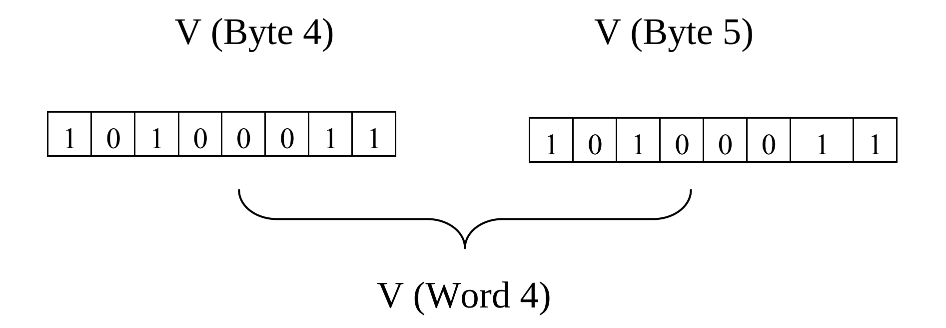
لكل وحدة من الذاكرة أسم و طريقة للكتابة:

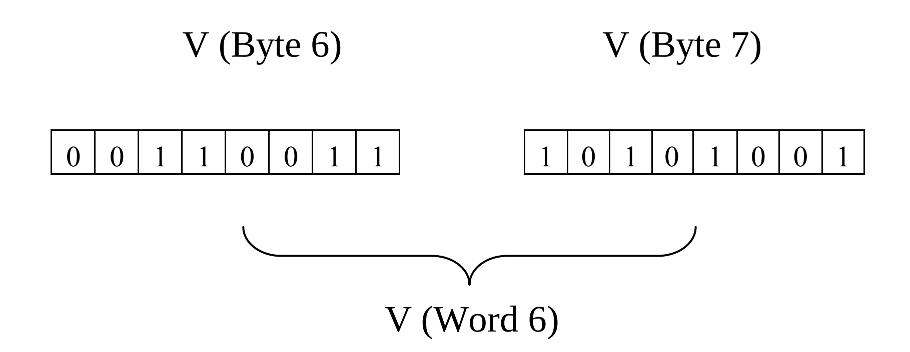
- VBIT0, VBIT 1, VBIT 2, VBIT 3, VBIT 4...................
- VBYTE 0, VBYTE 1, VBYTE 2, VBYTE 3, ...
- VWORD 0, VWORD 2, VWORD 4, VWORD 6, ...
- VD.WORD 0, VD.WORD 4, VD.WORD 8, VD.WORD 12

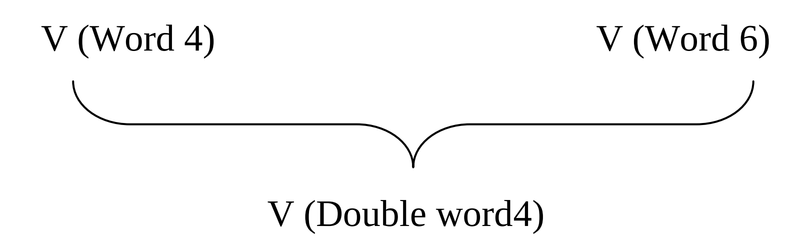
ملاحظة:

من المهم جداً مرعاه أن فى حالة كتابة معلومات بواسطة الــ bits يجب بدء الكتابة من اليمين إلى اليسار أما فى حالة كتابة معلومات بواسطة الــ bytes أو الــ words أو الــ D.words يجب بدء الكتابة من اليسار إلى اليمين وكذالك أيضاً فى حالة قراءة البيانات.

شرح مفصل:







لفهم طريقة القراءة أنظر صفحة 64

الأستخدامات:

يستخدم الـــ **V**bit مثل الريليه تماماً مثلاً:

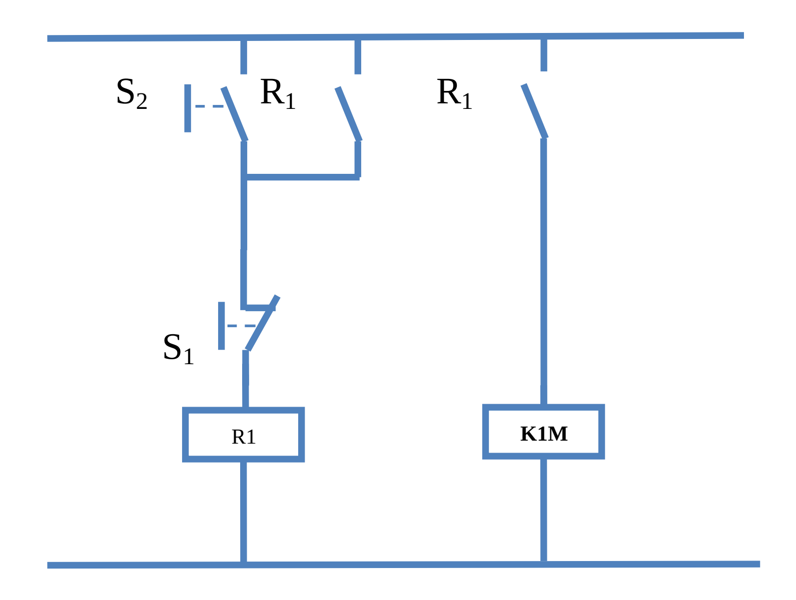
- V0.3 , V0.2 , V0.1 , V0.0

بينما يستخدم الـــ **V**Byte و **V**Word و **VD**.word كذاكرة لتخزين البيانات مثلاً:

- VB2 , VB1 , VB0
- VW4 , VW2 , VW0
- VD8 , VD4 , VD0

✓ مثال باستخدام الـ Vbit مثل الـريليه:

عدد الدخل	نوع الدخل	أسم الدخل
١	n.c.	I0.0/S1
٢	n.o.	I0.1/S2
عدد المتغيرات	نوع المتغيرات	أسم المتغيرات
١	bit	V0.0
عدد الخرج	نوع الخرج	أسم الخرج
١	كونتكتور	Q0.2/K1M



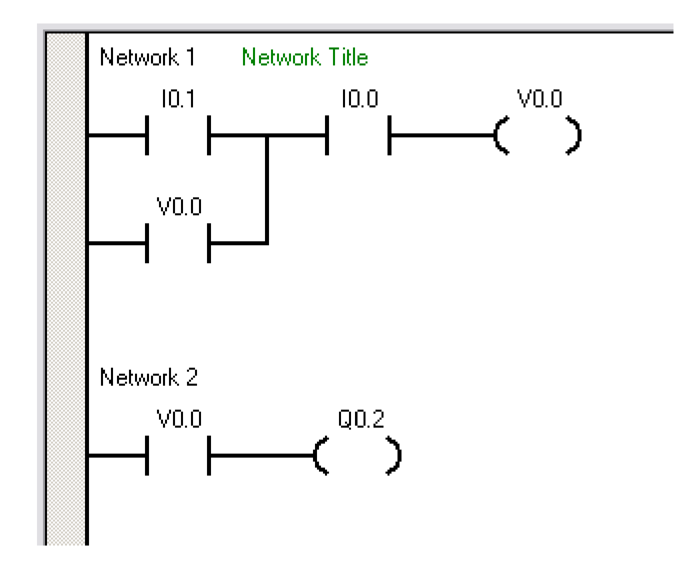
Network1:

بالضغط على I0.1 يعمل الـ V0.0 مثل الـ MARKER أو الـ RELAY.

Network2:

عندما يعمل الـ V0.0 يعمل أيضاً المحرك Q0.2

البرنامج:



- مثال بأستخدام الـــ Vword:

يمكن استخدام المتغيرات مع المؤقتات الزمنية كما بالتمرين التالى:

✓ عندما يصل المؤقت الزمنى إلى وقت معين يجب أن يقوم بفصل لمبة و أضاءه الأخرى بشرط أن يكون الزمن قابل للتغيير.

عدد الدخل	**نوع الدخل**	**أسم الدخل**
١	n.o.	I0.0/S1
عدد المتغيرات	**نوع المتغيرات**	**أسم المتغيرات**
١	word	VW0
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتكتور	Q0.0/K1M
٢	كونتكتور	Q0.1/K2M

الشرح:

:Network1

بالضغط على I0.0 يبدأ المؤقت الزمنى T200 بالعمل.

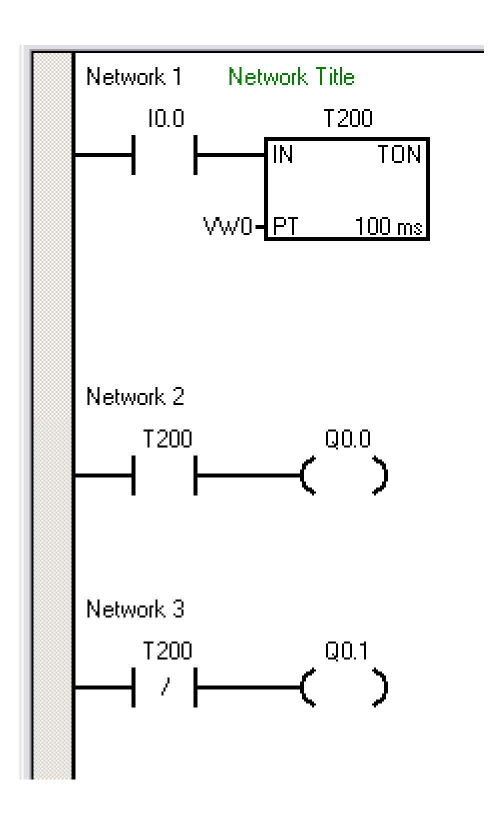
:Network2

عندما يصل المؤقت الزمنى إلى الوقت المحدد بواسطة المتغيرات يقوم بتشغيل الخرج Q0.0

:Network3

عندما يصل المؤقت الزمنى إلى الوقت المحدد بواسطة المتغيرات يقوم بفصل الخرج Q0.1

البرنامج:



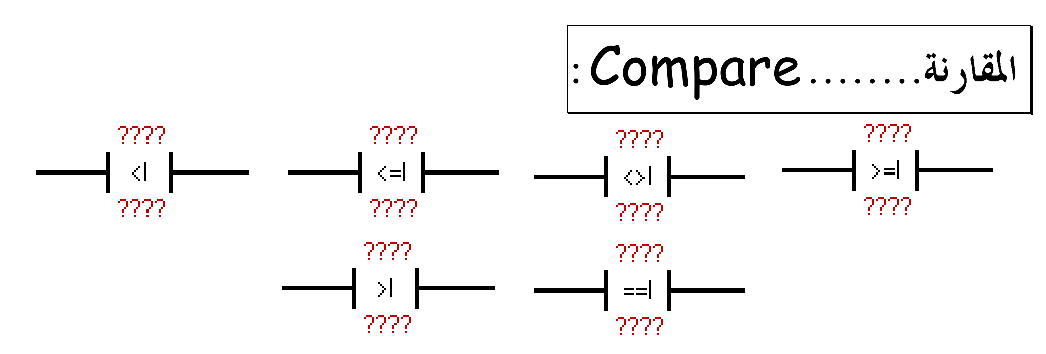
ملاحظة:

- فى البداية تكون قيمة المتغيرات صفر وهذا يسبب مشكلة كبيرة فى هذا التمرين لأنه فى هذا التمرين سيقوم المؤقت الزمنى بتغير النقاط الخاصة به عندما يصل إلى صفر أى فى نفس اللحظة التى سيغلق فيها المفتاح I0.0 فسيبدو أنه يعمل مثل الريليه تماماً.
- لحل هذه المشكلة يتم استخدام صفحة المتغيرات لتحديد قيمة مسبقة لأى من المتغيرات المستخدمة فى البرنامج.
- القيمة المسبقة التى تحدد بواسطة صفحة المتغيرات قابلة للتغير أثناء تنفيذ البرنامج كما سنلاحظ فيما بعد "الجزء الثانى".

الباب الثامن

المقارنة

- أنواع مفاتيح المقـــــــــــــارنة داخل الــ PLC.
- مفاتيح المقـــــــــــــارنة داخل الــ CPU 214.
- مفاتيح المقـــــــــــــارنة داخل الــ CPU 224.
- مفاتيح المقـــــــــــــــــــارنة نوع Byte.
- مفاتيح المقـــــــــــــــــــارنة نوع Word.
- مفاتيح المقـــــــــــــــــــارنة نوع Dword.
- مسميات مفاتيح المقـــــــــــــــــــارنة.
- تماريـــــــــن تطبيقية بأستخدام مفاتيح المقـــــارنة.



كل مفتاح من مفاتيح المقارنة هو عبارة عن معادلة بحيث أنة عندما تتحقق هذه المعادلة يصبح المفتاح مغلق في ماعدا ذالك يبقى المفتاح مفتوح.

كل مفاتيح المقارنة توجد في الــ CPU224 ولكن ليست جميعها توجد في الــ CPU214.

م	CPU 214	CPU 224	مفتاح المقارنة
١	√	√	???? ==I ????
٢	---------	√	???? <>I ????
٣	√	√	???? >=I ????
٤	---------	√	???? >I ????
٥	√	√	???? <=I ????
٦	---------	√	???? <I ????

كما أنه سيتم شرح كيفية قراءة المعادلة الخاصة بكل مفتاح من مفاتيح المقارنة على حدا فى الجدول التالى.

م	الأسم	الشرح	الشكل
١	**Equal to** **يساوى**	عندما تتساوى القيمة المكتوبة فوق المفتاح مع القيمة المكتوبة أسفل المفتاح و العكس صحيح يصبح المفتاح مغلق فى ماعدا ذالك يبقى المفتاح مفتوح.	???? ==| ????
٢	**Greater than or equal** **أكبر من أو يساوى**	عندما تصبح القيمة المكتوبة فوق المفتاح أكبر من أو تساوى القيمة المكتوبة أسفل المفتاح و العكس صحيح أى أن القيمة المكتوبة أسفل المفتاح أصغر من أو تساوى القيمة المكتوبة فوق المفتاح فأن المفتاح يصبح مغلق فى ماعدا ذالك يبقى المفتاح مفتوح.	???? >=| ????
٣		عندما تصبح القيمة المكتوبة فوق المفتاح أصغر من	

	Less than or equal **أصغر من أو يساوى**	أو تساوى القيمة المكتوبة أسفل المفتاح و العكس صحيح أى أن القيمة المكتوبة أسفل المفتاح أكبر من أو تساوى القيمة المكتوبة فوق المفتاح فأن المفتاح يصبح مغلق فى ماعدا ذالك يبقى المفتاح مفتوح.	???? <=I ????
٤	**Not equal to** **لا يساوى**	عندما لا تتساوى القيمة المكتوبة فوق المفتاح مع القيمة المكتوبة أسفل المفتاح و العكس صحيح يصبح المفتاح مغلق فى ماعدا ذالك يبقى المفتاح مفتوح.	???? <>I ????
٥	**Greater than** **أكبر من**	عندما تصبح القيمة المكتوبة فوق المفتاح أكبر من القيمة المكتوبة أسفل المفتاح و العكس صحيح أى أن القيمة المكتوبة أسفل المفتاح أصغر من القيمة المكتوبة فوق المفتاح فأن المفتاح يصبح مغلق فى ماعدا ذالك يبقى المفتاح مفتوح.	???? >I ????

٦	**Less than** **أصغر من**	عندما تصبح القيمة المكتوبة فوق المفتاح أصغر من القيمة المكتوبة أسفل المفتاح و العكس صحيح أى أن القيمة المكتوبة أسفل المفتاح أكبر من القيمة المكتوبة فوق المفتاح فأن المفتاح يصبح مغلق فى ماعدا ذالك يبقى المفتاح مفتوح.	???? <I ????

أنواع المقارنات الموجودة بجهاز الـ PLC:

	م	الأنواع	توضيح	الشكل
أنواع المقارنات	١	Byte	يستخدم لمقارنة Byte مع Byte ولهذا أكبر رقم يمكن كتابته هو ٢٥٥	???? ==B ????
	٢	Word	يستخدم لمقارنة Word مع Word ولهذا أكبر رقم موجب يمكن كتابته هو +٣٢٧٦٧ بينما أكبر رقم سالب هو -٣٢٧٦٨	???? ==I ????
	٣	D.Word	يستخدم لمقارنة D.Word مع D.Word ولهذا أكبر رقم يمكن كتابته هو ٤٢٩٤٩٦٧٢٩٥	???? ==D ????

أنواع المقارنات	٤	Real	يستخدم لمقارنة D.Word مع D.Word وهذا لأن الأرقام التي هي بالعلامة العشرية تكتب فقط على D.Word	???? ==R ????
	٥	String	يستخدم لمقارنة Byte مع Byte وهذا لأن الحرف الواحد يكتب فقط على Byte.	???? ==S ????

كل ما يمكن كتابته فوق أو تحت مفتاح المقارنة:

م	النوع	المكان	الرموز المستخدمة
١	???? ==B ????	مسميات فوق المفتاح	ثوابت , IB , QB , MB , VB , SMB , AC
		مسميات أسفل المفتاح	ثوابت , IB , QB , MB , VB , SMB , AC
٢	???? ==I ????	مسميات فوق المفتاح	ثوابت , IW , QW , MW , VW , SMW , AC , T , C
		مسميات أسفل المفتاح	ثوابت , IW , QW , MW , VW , SMW , AC , T , C , AIW , AQW

٣	???? ==D ????	مسميات فوق المفتاح	ثوابت , ID , QD , MD , VD , SMD , AC
		مسميات أسفل المفتاح	ثوابت , ID , QD , MD , VD , SMD , AC
٤	???? ==R ????	مسميات فوق المفتاح	ثوابت , ID , QD , MD , VD , SMD , AC
		مسميات أسفل المفتاح	ثوابت , ID , QD , MD , VD , SMD , AC
٥	???? ==S ????	مسميات فوق المفتاح	ثوابت , IB , QB , MB , VB , SMB , AC
		مسميات أسفل المفتاح	ثوابت , IB , QB , MB , VB , SMB , AC

شرح كل ما يمكن كتابته على مفاتيح المقارنة:

م	الأنواع	التوضيح
١	ثوابت	قيم تتم كتابتها أثناء البرمجة وهى غير قابلة للتغير أثناء عمل البرنامج.
٢	IB	هى عبارة عن مجموعة من ثمان مفاتيح.
٣	IW	هى عبارة عن مجموعة من ستة عشر مفتاح.
٤	ID	هى عبارة عن مجموعة من أثنين وثلاثين مفتاح.
٥	QB	هى عبارة عن مجموعة من ثمان مخرجات.

٦	QW	هى عبارة عن مجموعة من ستة عشر خرج.
٧	QD	هى عبارة عن مجموعة من أثنين وثلاثين خرج.
٨	MB	هى عبارة عن مجموعة من ثمان ريلهات.
٩	MW	هى عبارة عن مجموعة من ستة عشر ريليه.
١٠	MD	هى عبارة عن مجموعة من أثنين وثلاثين ريليه.
١١	VB	هى عبارة عن متغيرات بحجم byte.
١٢	VW	هى عبارة عن متغيرات بحجم word.
١٣	VD	هى عبارة عن متغيرات بحجم Dword.
١٤	SMB	هى عبارة عن مجموعة من ثمان ريليهات خاصة.
١٥	SMW	هى عبارة عن مجموعة من ستة عشر ريليه خاص.
١٦	SMD	هى عبارة عن مجموعة من أثنين وثلاثين ريليه خاص.
١٧	AC	هو عبارة عن محتوى للقيام بالعمليات الحسابية.

أمثلة عملية:

١- قم بتنفيذ دائرة تحكم منطقية لمحركين بحيث المحرك الأول يعمل بعد ثلاث ثوانى من ضغط المفتاح بينما يعمل المحرك الثانى بعد خمس ثوانى من الضغط على نفس المفتاح باستخدام مؤقت زمنى واحد.

٢- قم بتنفيذ دائرة تحكم منطقية للمبتان فلاشر بحيث يعملان بالتبادل فتعمل اللمبة الأولى لمدة ثانية بنما تبقى الأخرى مطفأة لمدة ثانية وهكذا.

٣- قم بتنفيذ دائرة تحكم منطقية لمحرك يعمل لليمين لزمن ثم يقف زمن أخر و يعمل يساراً لزمن ثم يقف لزمن أخر وهكذا.

المثال الأول:

- ✓ لمحركين بحيث المحرك الأول يعمل بعد ثلاث ثواني من ضغط المفتاح بينما يعمل المحرك الثاني بعد خمس ثواني من لضغط على نفس المفتاح باستخدام مؤقت زمني واحد.

عدد الدخل	نوع الدخل	أسم الدخل
١	n.c.	I0.0/S1
٢	n.o.	I0.1/S1
عدد مفاتيح المقارنة	**نوع مفاتيح المقارنة**	**أسم مفاتيح المقارنة**
١	==I	T50
٢	==I	T50
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتاكتور	Q0.4/K1M
٢	كونتاكتور	Q0.7/K1M

الشرح:

:Network1

يالضغط على I0.1 يبدأ المؤقت الزمني T50 بالعمل.

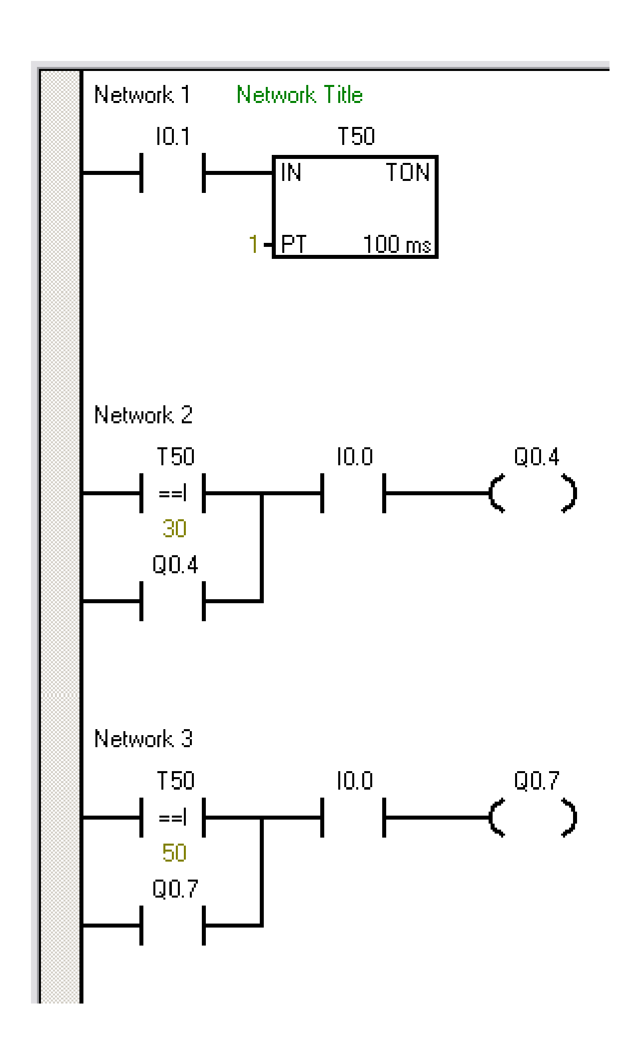
:Network2

عندما يصل المؤقت الزمني إلى ثلاث ثواني يغلق مفتاح المقارنة فيعمل المحرك Q0.4

:Network3

عندما يصل المؤقت الزمني إلى خمس ثواني يغلق مفتاح المقارنة فيعمل المحرك Q0.7

البرنامج:


Network 1
Network Title
I0.1
T50
IN
TON
1
PT
100 ms
Network 2
T50
==I
30
I0.0
Q0.4
Q0.4
Network 3
T50
==I
50
I0.0
Q0.7
Q0.7

المثال الثاني:

- ✓ لمبتان فلاشر بحيث يعملان بالتبادل فتعمل اللمبة الأولى لمدة ثانية بنما تبقى الأخرى مطفأة لمدة ثانية وهكذا.

عدد الدخل	نوع الدخل	أسم الدخل
١	n.c.	I0.1/S1
عدد مفاتيح المقارنة	**نوع مفاتيح المقارنة**	**أسم مفاتيح المقارنة**
١	<=I	T33
٢	<=I	T33
٣	>=I	T33
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	لمبة	Q0.4/K1M
٢	لمبة	Q0.7/K1M

الشرح:

:Network1

يالضغط على I0.1 يبدأ المؤقت الزمنى T33 بالعمل.

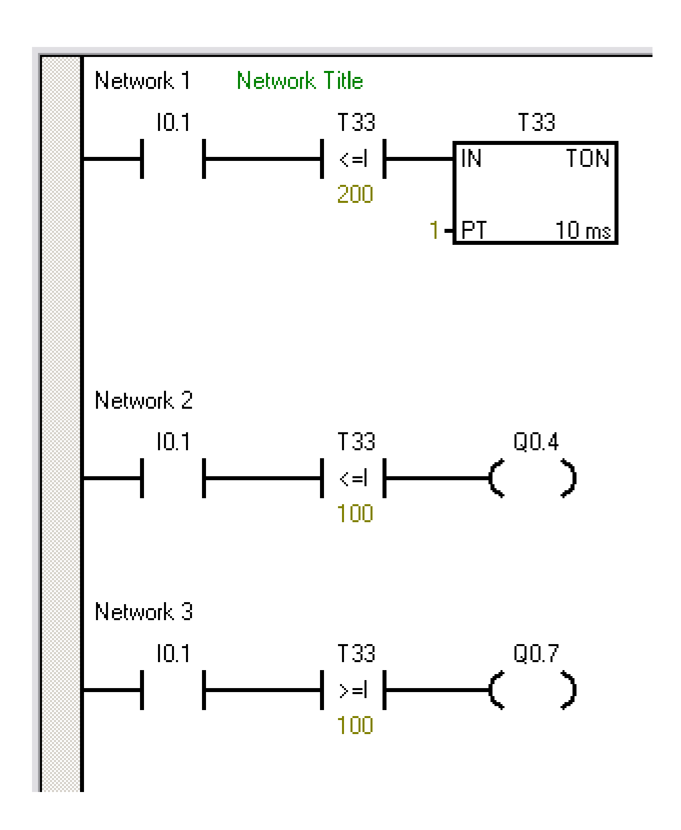
:Network2

بالضغط على I0.1 تبدأ Q0.4 بالعمل حتى أن يتعدى المؤقت الزمنى الثانية.

:Network3

تبدأ Q0.7 بالعمل عندما يتعدى المؤقت الزمنى ثانية وعندا يصل المؤقت الزمنى إلى ثانيتان يتوقف المؤقت الزمنى ليبدأ من جديد.

البرنامج:


Network 1
Network Title
I0.1
T33
<=I
200
T33
IN
TON
1
PT
10 ms
Network 2
I0.1
T33
<=I
100
Q0.4
Network 3
I0.1
T33
>=I
100
Q0.7

المثال الثالث:

✓ لمحرك يعمل لليمين لزمن ثم يقف زمن أخر و يعمل يساراً لزمن ثم يقف لزمن أخر وهكذا.

عدد الدخل	نوع الدخل	أسم الدخل
١	n.o.	I1.1/S1
عدد مفاتيح المقارنة	**نوع مفاتيح المقارنة**	**أسم مفاتيح المقارنة**
١	>=I	T32
٢	<=I	T32
٣	<=I	T32
٤	<=I	T32
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتاكتور	Q0.2/K1M
٢	كونتاكتور	Q3.1/K1M

الشرح:

:Network1

بالضغط على I1.1 يبدأ المؤقت الزمنى T32 بالعمل.

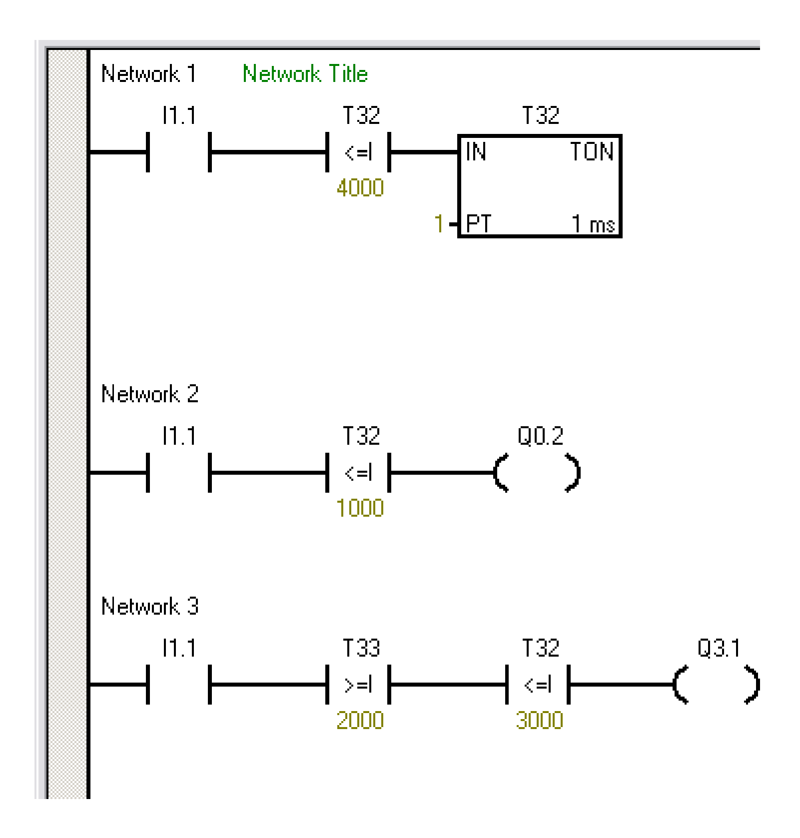
:Network2

بالضغط على I1.1 تبدأ Q0.2 بالعمل حتى أن يتعدى المؤقت الزمنى الثانية.

Network3:

تبدأ Q3.1 بالعمل عندما يتعدى المؤقت الزمني الثانيتان وعندا يصل المؤقت الزمني إلى ثلاث ثوانى يتوقف المؤقت الزمني لمدة ثانية أخرى ليبدأ من جديد.

البرنامج:

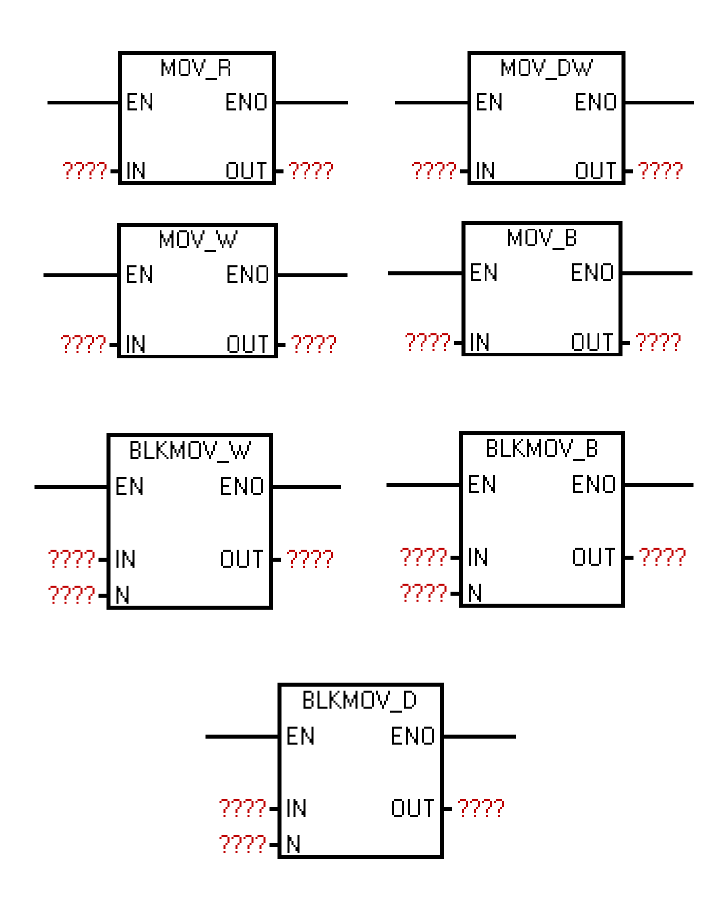


الباب التاسع

عمليات النقل

- أنواع عمليات النقـــــــــــل داخل الــ PLC.
- عمليات نقـــل للأرقام الصحيحة و للأرقام العشرية.
- عمليات نقـــــــــــــــل لحجم الــ byte.
- عمليات نقـــــــــــل مجموعات بحجم الــ byte.
- عمليات نقـــــــــــــــل لحجم الــ word.
- عمليات نقـــــــــــل مجموعات بحجم الــ word.
- عمليات نقل لحجم الــ Dword للأرقام الصحيحة.
- عمليات نقـــل لحجم الــ Dword للأرقام العشرية.
- عمليات نقـــــــــــل مجموعات بحجم الــ Dword.
- تماريـــــن تـــطـــبـــيـــقـــيـــة على عمليات النقل.

عمليات نقل القيم.......MOVE :



- تستخدم عمليات نقل القيم MOVE لنقل أى قيمة من داخل أى ذاكرة إلى أى ذاكرة أخرى مع مراعاة أن يكون حجم الذاكرة التى سوف يتم نقل القيمة لها هو نفس حجم للذاكرة التى تم نقل القيمة منها.

- تستخدم عمليات نقل المجموعات BLKMOVE لنقل أى عدد من Byte أو Word أو D.word مع مراعاة أن يكون حجم الذاكرة التي سوف يتم نقل القيمة لها هو نفس حجم الذاكرة التي تم نقل القيمة منها.
- يوجد سهم على يمين الـــ MOVE أو BLKMOVE يعمل كمفتاح يغلق عندما يتم تنفيذ العملية المراده.

أنواع عمليات النقل:

م	الأسم	الشرح	الشكل
١	عمليات النقل Byte	عمليات نقل الـــ Byte تستخدم فى نقل أى ثوابت (أرقام صحيحة) أو متغيرات من مشتقات الـــ Byte إلى أى متغيرات أخرى من مشتقات الـــ Byte.	MOV_B EN ENO ???? IN OUT ????
٢	عمليات النقل Word	عمليات نقل الـــ Word تستخدم فى نقل أى ثوابت (أرقام صحيحة فقط) أو متغيرات من	

		مشـــــتقات الــــــ Word إلى أى متغيرات أخرى من مشتقات الـــــ Word.	MOV_W EN ENO ???? IN OUT ????
٣	عمليات النقل Dword	عمليات نقــل الـــــ D.Word تستخدم فى نقل أى ثوابت (أرقـــام صحيحة فقط) أو مـــتغيرات مـــن مشـــــتقات الـــــ D.Word إلى أى متغيرات أخرى مـــن مشـــتقات الـــ D.Word.	MOV_DW EN ENO ???? IN OUT ????
٤	عمليات النقل Real	عمليات نقــل الـــــ D.Word تستخدم فى نقل أى ثوابت (أرقـــام عشرية فقـــط) أو مـــتغيرات مـــن مشـــــتقات الـــــ D.Word إلى أى متغيرات أخرى مـــن مشـــتقات الـــ D.Word.	MOV_R EN ENO ???? IN OUT ????

أنواع عمليات نقل المجموعات:

م	الأسم	الشرح	الشكل
١	عمليات نقل لمجموعة Byte	عمليات نقل لمجموعة الـــــ Byte تســـتخدم فى نقل أى عدد من الــــ Byte بشـــرط أن تكون متتالية فى الترتيـــب. يمكـــن أن تكـــون مجموعة الــ Byte تحتـــوى علـــى ثوابـــت (أرقام صحيحة فقط) أو أى مـــتغيرات مـــن مشـــتقات الـــــ Byte إلى أى مـــتغيرات أخرى من مشتقات الــ Byte.	BLKMOV_B EN ENO ???? IN OUT ???? ???? N
٢	عمليات نقل لمجموعة Word	عمليـــات نقـــل لمجموعـــة الـــــ Word تستخدم فى نقل أى عدد مـــن الـــــ Word بشرط أن تكون متتالية فى الترتيـــب. يمكـــن أن تكون مجموعـــة الـــــ Word تحتـــوى	BLKMOV_W EN ENO ???? IN OUT ???? ???? N

		على ثوابت (أرقام صحيحة فقط) أو أى متغيرات من مشتقات الـ Word إلى أى متغيرات أخرى من مشتقات الـ Word.	
٣	عمليات نقل لمجموعة Dword	عمليات نقل لمجموعة الـ DWord تستخدم فى نقل أى عدد من الـ DWord بشرط أن تكون متتالية فى الترتيب. يمكن أن تكون مجموعة الـ DWord تحتوى على ثوابت (أرقام صحيحة أو أرقام عشرية) أو أى متغيرات من مشتقات الـ DWord إلى أى متغيرات أخرى من مشتقات الـ DWord.	BLKMOV_D EN ENO ??? IN OUT ???? ??? N

كل ما يمكن كتابته فى دخل أو خرج عمليات النقل:

م	النوع	المكان	الرموز المستخدمة
١	MOV_B EN ENO ???? IN OUT ????	مسميات الدخل IN	ثوابت(أرقام صحيحة فقط) , IB , QB , AC , SMB , VB , MB
		مسميات الخرج OUT	QB AC , SMB , VB , MB ,
٢	MOV_W EN ENO ???? IN OUT ????	مسميات الدخل IN	ثوابت(أرقام صحيحة فقط) , IW , QW , AC , SMW , VW , MW
		مسميات الخرج OUT	, SMW , VW , MW , QW AC
٣	MOV_DW EN ENO ???? IN OUT ????	مسميات الدخل IN	ثوابت(أرقام صحيحة فقط) , ID , QD , AC , SMD , VD , MD
		مسميات الخرج OUT	QD AC , SMD , VD , MD ,
٤	MOV_R EN ENO ???? IN OUT ????	مسميات الدخل IN	ثوابت(أرقام عشرية فقط) , ID , QD , AC , SMD , VD , MD
		مسميات الخرج OUT	QD AC , SMD , VD , MD ,

كل ما يمكن كتابته فى دخل أو خرج عمليات نقل المجموعات:

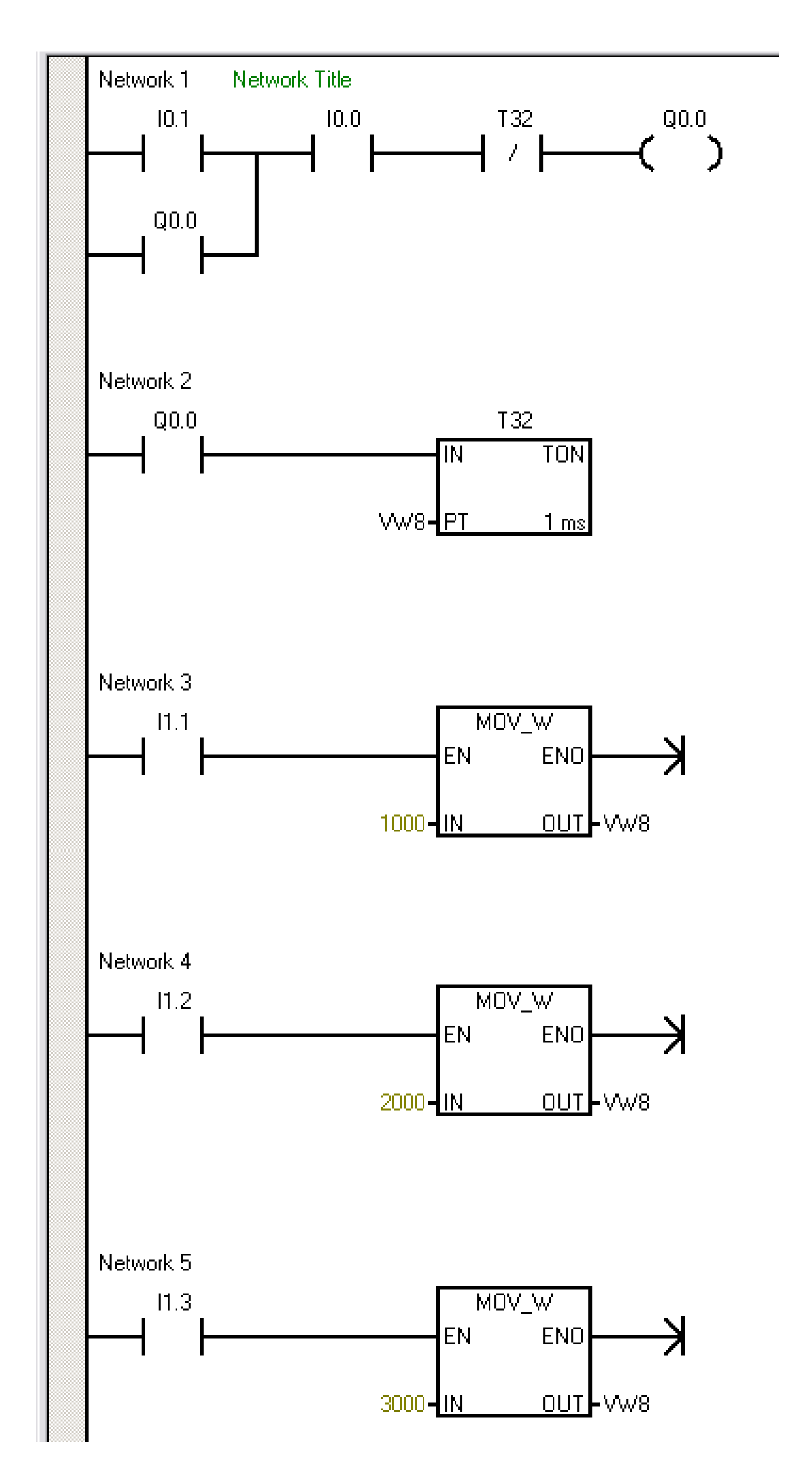
م	النوع	المكان	الرموز المستخدمة
١	BLKMOV_B EN ENO ????-IN OUT-???? ????-N	مسميات الدخل IN	IB , QB , MB , VB , SMB , AC
		مسميات الخرج OUT	QB , MB , VB , SMB , AC
		N	من ١ إلى ٢٥٥
٢	BLKMOV_W EN ENO ????-IN OUT-???? ????-N	مسميات الدخل IN	IW , QW , MW , VW , SMW , AC
		مسميات الخرج OUT	QW , MW , VW , SMW , AC
		N	من ١ إلى ٢٥٥
٣	BLKMOV_D EN ENO ????-IN OUT-???? ????-N	مسميات الدخل IN	ID , QD , MD , VD , SMD , AC
		مسميات الخرج OUT	QD , MD , VD , SMD , AC
		N	من ١ إلى ٢٥٥

أمثلة (تمارين عملية):

١– قم بتنفيذ دائرة تحكم منطقية لمحرك يعمل لمدة زمن ثم يقف, بشرط أن يكون هذا الزمن متغير (١ثانية – ٢ثانية – ٣ثانية).

عدد الدخل	**نوع الدخل**	**أسم الدخل**
١	n.c.	I0.0/S1
٢	n.o.	I0.1/S2
٣	n.o.	I1.1/S3
٤	n.o.	I1.2/S4
٥	n.o.	I1.3/S5
عدد المؤقتات الزمنية	**نوع المؤقتات الزمنية**	**أسم المؤقتات الزمنية**
١	TON	T32
عدد عمليات النقل	**نوع عمليات النقل**	**أسم عمليات النقل**
١	Word	MOV_W
٢	Word	MOV_W
٣	Word	MOV_W
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتكتور	Q0.0/K1M

التمرين:


Network 1
Network Title
I0.1
I0.0
T32
Q0.0
Q0.0
Network 2
Q0.0
T32
IN
TON
VW8
PT
1 ms
Network 3
I1.1
MOV_W
EN
ENO
1000
IN
OUT
VW8
Network 4
I1.2
MOV_W
EN
ENO
2000
IN
OUT
VW8
Network 5
I1.3
MOV_W
EN
ENO
3000
IN
OUT
VW8

الشرح:

:Network1

بالضغط على I0.1 يعمل الخرج Q0.0 إلى أن يتم فصلة بواسطة مفتاح الإيقاف أو بواسطة النقطة المغلقة الخاصة بالمؤقت الزمنى T32.

:Network2

عندما يعمل الخرج يبدأ المؤقت الزمنى T32 بالعمل.

:Network3

بالضغط على I1.1 تصبح قيمة الــ VW8 هى ١٠٠٠ فيعمل المؤقت الزمنى لثانية واحدة.

:Network4

بالضغط على I1.2 تصبح قيمة الــ VW8 هى ٢٠٠٠ فيعمل المؤقت الزمنى لثانيتان.

:Network5

بالضغط على I1.2 تصبح قيمة الــ VW8 هى ٣٠٠٠ فيعمل المؤقت الزمنى لثلاث ثوانى.

الفهرس

الفهرس

الفهرس

الفهرس

الكتب التي صدرت عن معهد السالزيان الإيطالى
"دون بوسكو"

🕮 **محركات, مولدات و محولات التيار المتردد**	وجيه جرجس
🕮 **دوائر التحكم الآلي** الجزء الأول	وجيه جرجس
🕮 **دوائر التحكم الآلي** الجزء الثاني	وجيه جرجس
🕮 **الغسالة الفول أوتوماتك** الجزء الأول	وجيه جرجس
🕮 **الغسالة الفول أوتوماتك** الجزء الثاني	وجيه جرجس
🕮 **الدوائر العملية للضغوط الهوائية و الكهروهوائية**	وجيه جرجس
🕮 **غسالة الأطباق**	وجيه جرجس
🕮 **زانوسى الموديلات القديمة 14-16-18برروجرام**	وجيه جرجس
🕮 **الدوائر الكهربائية للتركيبات المنزلية**	نبيل رزق
🕮 **صيانة وإصلاح الأجهزة المنزلية**	نبيل رزق
🕮 **أفكار التكيف و التبريد للدوائر الميكانيكية**	إميل فتح الله
🕮 **أفكار التكيف و التبريد للدوائر الكهربائية**	إميل فتح الله
🕮 **أفكار التكيف و التبريد الخدمة والأعطال**	إميل فتح الله
🕮 **برمجة التحكم المنطقي .P.L.C** الجزء الأول	ريمون كمال
🕮 **برمجة التحكم المنطقي .P.L.C** الجزء الثاني	تحت التحضير
🕮 **موديلات الغسالة كريازى**	تحت التحضير

برمجة التحكم المنطقى .P.L.C

الجزء الثانى

إعداد

ريمون كمال

معهد السالزيان الإيطالى "**دون بوسكو**"

٢شارع عبد القادر طه – الساحل   ت: ٢٤٥٧٩٦٥٠ – ٢٤٥٧٦٧٩٤

معهد فنى – معهد صناعى

دورات تدريبية سريعة مركزة

دورات تدريبية تعليمية للمدرسين

المراجع

١. أجهزة التحكم المبرمج وتطبيقاتها العملي.

1. Controllore a logica programmabile P. Bani.
2. Siemens Programmable Controller Manual.

طبعة جديدة

2012

أسم الكتاب:	**برمجة التحكم المنطقى *P.L.C.***
	الجزء الثانى
طباعة	
رقم الإيداع:	
الترقيم الدولى:	

شكر و إهداء

أهدى هذا الكتاب إلى أبي وأمي الذين لهم كل الفضل بأن أعمل في هذا المجال وهم الذين شجعوني على عمل هذا الكتاب بكل جهد وإخلاص شاكر الله و إياهم وكل من ساهم في تقديمه.

وأشكر أيضاً كل المعلمين الأفاضل الذين ساعدوا على خروج هذا الكتاب إلى الملىء.

- **المدير الإيطالي للمعهد:** الأب رينسو ليوناردوسى
- **الناظر السابق للمعهد:** الأب بيرناردو أشيربونى
- **مدير الدورات التدريبية:** أ. ماجد چورچ
- **أستاذ التحكم:** أ. نبيل رزق **–** أ. وجية جرجس
- **أستاذ التكيف والتبريد:** أ. إميل فتح الله
- **أستاذ الـــ PLC:** أ. ماجد موريس **–** أ. ماجد عريان **–** أ. چيوليو جالو **–** أ. محسن أنطون

أشكر كل من أرسلوا لى التعليقات بخصوص الجزء الأول من الكتاب وقد حاولت قدر المستطاع تلبية متطلباتهم فى الجزء الثانى وأتشرف باستقبال المزيد من تعليقات السادة القراء بخصوص هذا لجزء من الكتاب على عنوان البريد التالى

plcbook@hotmail.com

مقدمة

نظراً للتقدم العلمى السريع المرتبط بالمجال الصناعى وخاصة من الناحية الكهربية أصبح لا غنى عن الربط بين عالم الصناعة وبين التكنولوچيا العصرية ويتمثل هذا الربط بواسطة استخدام أجهزة التحكم المنطقى بمختلف أنواعها والتى تستحق أن تسمى بالأجهزة الذكية نظراً لما تقدمه فى المجال الصناعى من: سهولة فى تصميم البرامج، ومرونة فى أكتشاف الأعطال، ومساعدة فى حل المشاكل، ...الخ

و نظراً لصعوبة ترجمة بعض المصطلحات الخاصة بهذا المجال وخاصة لكى لا تفقد المعنى التقنى أو الفنى لها، تمت كتابتها بلغتها الأصلية لذلك لا تهتم كثيراً عزيزى القارئ بهذه المصطلحات فستكون بسيطة ومفهومة بمجرد ما أن تتعمق بفهم فى هذا المجال.

هكذا أيضاً لا تتعجل عزيزى القارئ فى النظر إلى مواضيع متباعدة خاصة أن كنت بمبتدئ فى هذا المجال وهذا لأنه قد تم شرح المنهج بطريقة متسلسلة ولذلك يفضل للقارئ قراءة المواضيع بالتسلسل التى كتبت به لفهم جميع الأمور دون تخبط.

و خاصاً لفهم التمارين لا يشترط فقط القراءة بترتيب بل يجب أيضاً أن تربط كل شرح و كل رمز بالرسم الموجود ولا تقوم بالقراءة بطريقة عابرة.

تم شرح البرمجة بطريقة عامة دون اللجؤ إلى ماركة بعينها وهذا لكى يخدم كل من يعمل مع وحدات التحكم المنطقى بمختلف أنواعها.

تنقسم معرفة أجهزة التحكم المنطقى إلى أمور عديدة من أهمها:

تصميم برامج – اكتشاف أعطال – حل مشاكل

قد تم التركيز بشكل كبير فى الجزء الأول من هذا الكتاب على معرفة أساسات تصميم البرامج بطريقة سلسة وباستخدام أسهل لغات البرمجة ولذلك أقدم لكم الجزء الثانى لاستكمال شرح تصميم البرامج باستخدام أساليب ذات أكثر تطور ويتم الآن تحضير جزء خاص "بأكتشاف الأعطال – بحل المشاكل وبتصميم التمارين والأفكار العملية".

لذلك أقدم لكم الجزء الثانى من هذا الكتاب لخدمة كل من يدرس أو يعمل فى هذا المجال و أتمنى من الله أن يجد كل من يقرأ هذا الكتاب نفعاً له.

المـــؤلـــف

الباب الأول

العمليات الحسابية
للأرقام الصحيحة و العشرية

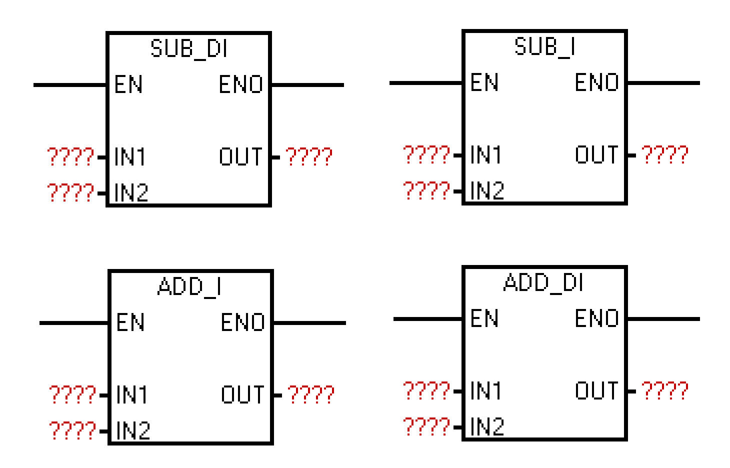
- أنواع العمـــــــــــــــــــــليات الحسابية.
- العمليــــــــــــــات الحسابية للأرقام الصحيحة.
- العمليــــــــــــــات الحسابية للأرقام العشرية.
- الفـــــــــــــــــــــــــــرق بين النوعين.
- طريقة توصيل العمليـــات الحسابيـــــــــــة.
- مـــلاحظات هامة بخصوص العمليات الحسابية.
- تمارين تطبيقية على العمليات الحسابيــــــة.

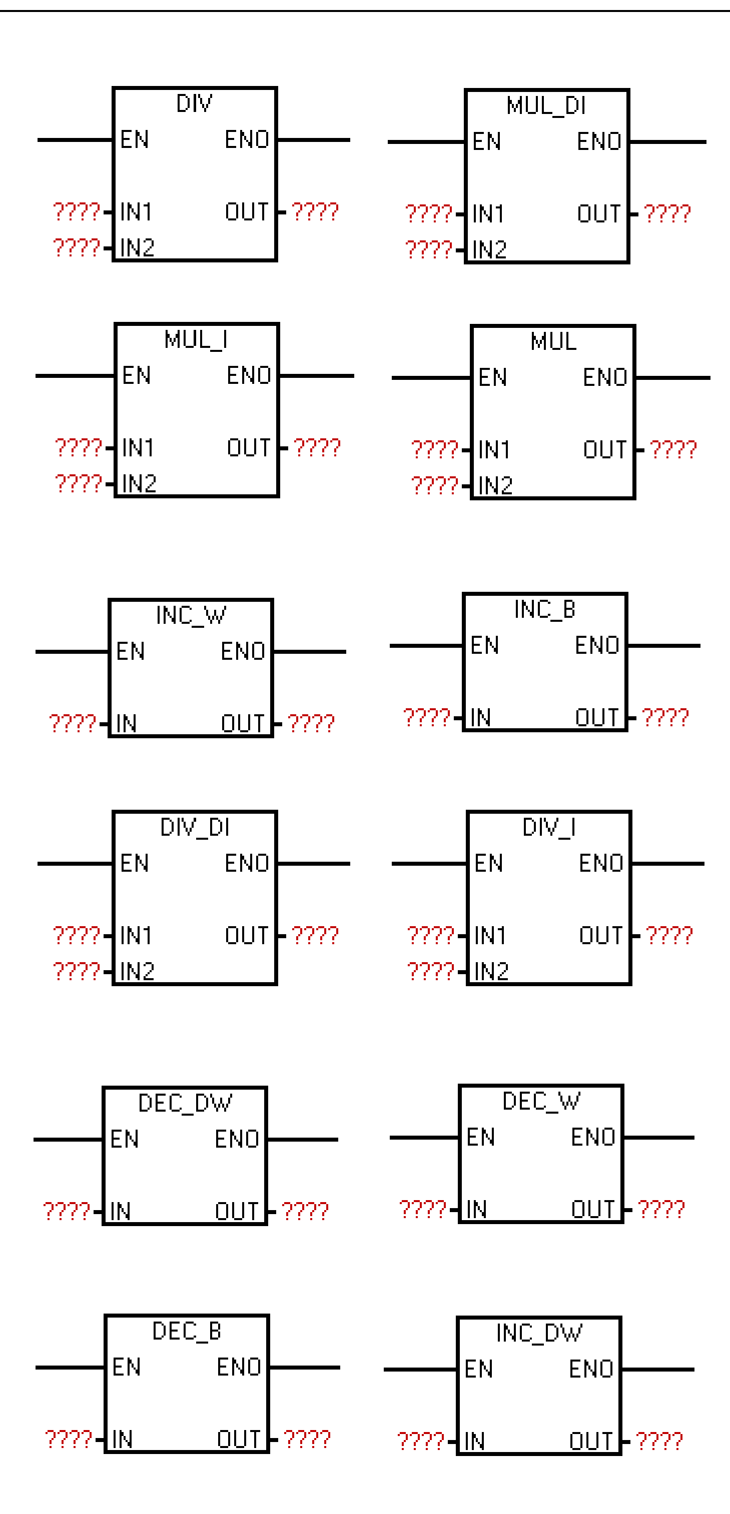
العمليات الحسابية:

تستخدم العمليات الحسابية فى بعض البرامج التى تحتوى على متغيرات أو التى يتم فيها التحويل من أى وحدة قياس إلى أى وحدة قياس مختلفة أو حتى للقيام بمعدلات من الدرجة الأولى, أو الثانية, أو ......

تنقسم العمليات الحسابية إلى نوعين, النوع الأول هو العمليات الحسابية للأرقام الصحيحة و الثانية هى العمليات الحسابية للأرقام العشرية.

العمليات الحسابية للأرقام الصحيحة:




DIV
EN
ENO
IN1
OUT
IN2
MUL_DI
EN
ENO
IN1
OUT
IN2
MUL_I
EN
ENO
IN1
OUT
IN2
MUL
EN
ENO
IN1
OUT
IN2
INC_W
EN
ENO
IN
OUT
INC_B
EN
ENO
IN
OUT
DIV_DI
EN
ENO
IN1
OUT
IN2
DIV_I
EN
ENO
IN1
OUT
IN2
DEC_DW
EN
ENO
IN
OUT
DEC_W
EN
ENO
IN
OUT
DEC_B
EN
ENO
IN
OUT
INC_DW
EN
ENO
IN
OUT
????

كل العمليات التى تتم بواسطة **العمليات الحسابية للأرقام الصحيحة** يجب أن تحتوى فقط على أرقام صحيحة و يجب أن تكون النواتج هى أيضاً صحيحة فمثلاً:

- **لا يمكن استخدام العمليات الحسابية الصحيحة لجمع أرقام غير صحيحة فمثلاً:**

لا يمكن جمع رقم بقيمة ٢,٥ و قيمة ٦,١ لأنها ليست أرقام صحيحة.

- **لا يمكن استخدام العمليات الحسابية الصحيحة لطرح أرقام غير صحيحة فمثلاً:**

لا يمكن طرح رقم بقيمة ٨,٥ من رقم ٦,١ لأنها ليست أرقام صحيحة.

- **لا يمكن استخدام العمليات الحسابية الصحيحة لقسمة أرقام غير صحيحة فمثلاً:**

لا يمكن قسمة رقم بقيمة ٠,١ و قيمة ٥,٣ لأنها ليست أرقام صحيحة.

- **لا يمكن استخدام العمليات الحسابية الصحيحة لضرب أرقام غير صحيحة فمثلاً:**

لا يمكن ضرب رقم بقيمة ٨,١٠ و قيمة ٣,٢ لأنها ليست أرقام صحيحة.

ملاحظة:

يجب أيضاً مراعاة الخرج بالنسبة للمعادلة أى مراعاة أن الناتج يجب أن يكون رقم صحيح و أن يكون الناتج موضوع على ذاكرة بالحجم الصحيح, فمثلاً:

- لا يمكن استخدام العمليات الحسابية الصحيحة بحجم byte لضرب قيمة ٢٠٠ و قيمة ٢٠٠ باستخدام MUL-B لأن الناتج لا يمكن أن يكتب على byte.

  لتجنب هذه المشكلة يتم التعامل مع عملية حسابية أخرى ذات حجم أكبر مثل: word.

- لا يمكن استخدام العمليات الحسابية الصحيحة لقسمة أرقام صحيحة مثل قيمة ١٠ و قيمة ٣ لأن الناتج لن سيكون رقم صحيح.
- لتجنب هذه المشكلة يتم التعامل مع عملية حسابية أخرى غير صحيحة كما سنوضح بعد قليل.

م	الاسم	الشرح	الشكل
١	ADD_I عملية جمع أرقام صحيحة بحجم Word.	عمليات جمع بحجم Word تقوم بجمع أرقام صحيحة (IN1 و IN2) ويكتب الناتج (OUT) على ذاكرة Word.	ADD_I EN ENO ????-IN1 OUT-???? ????-IN2
٢	ADD_DI عملية جمع أرقام صحيحة بحجم Dword.	عمليات جمع بحجم Dword تقوم بجمع أرقام صحيحة (IN1 و IN2) ويكتب الناتج (OUT) على ذاكرة Dword.	ADD_DI EN ENO ????-IN1 OUT-???? ????-IN2
٣	SUB_I عملية طرح أرقام صحيحة بحجم Word.	عمليات طرح بحجم Word تقوم بطرح أرقام صحيحة (IN2 من IN1) ويكتب الناتج OUT على ذاكرة Word.	SUB_I EN ENO ????-IN1 OUT-???? ????-IN2
٤	SUB_DI عملية طرح أرقام صحيحة بحجم Dword.	عمليات طرح بحجم Dword تقوم بطرح أرقام صحيحة (IN2 من IN1) ويكتب الناتج OUT على ذاكرة Dword.	SUB_DI EN ENO ????-IN1 OUT-???? ????-IN2

٥	MUL عملية ضرب أرقام صحيحة بحجم Dword/Word.	عمليات ضرب بحجم Word للمدخل و Dword للخرج، تقوم بضرب أرقام صحيحة بحجم Word (IN1 و IN2) ويكتب الناتج OUT على ذاكرة Dword.	MUL EN ENO ????-IN1 OUT-???? ????-IN2
٦	MUL_I عملية ضرب أرقام صحيحة بحجم Word.	عمليات ضرب بحجم Word تقوم بضرب أرقام صحيحة (IN1 و IN2) ويكتب الناتج OUT على ذاكرة Word.	MUL_I EN ENO ????-IN1 OUT-???? ????-IN2
٧	MUL_DI عملية ضرب أرقام صحيحة بحجم Dword.	عمليات ضرب بحجم Dword تقوم بضرب أرقام صحيحة (IN1 و IN2) ويكتب الناتج OUT على ذاكرة Dword.	MUL_DI EN ENO ????-IN1 OUT-???? ????-IN2
٨	DIV عملية قسمة أرقام صحيحة بحجم Dword/Word.	عمليات قسمة بحجم Word للمدخل و Dword للخرج، تقوم بقسمة أرقام صحيحة بحجم Word (IN1 على IN2) ويكتب الناتج OUT على ذاكرة Dword.	DIV EN ENO ????-IN1 OUT-???? ????-IN2

٩	DIV_I عملية قسمة أرقام صحيحة بحجم Word.	عمليات قسمة بحجم Word تقوم بقسمة أرقام صحيحة (IN1 على IN2) ويكتب الناتج OUT على ذاكرة Word.	DIV_I EN ENO ????-IN1 OUT-???? ????-IN2
١٠	DIV_DI عملية قسمة أرقام صحيحة بحجم Dword.	عمليات قسمة بحجم Word تقوم بقسمة أرقام صحيحة (IN1 على IN2) ويكتب الناتج OUT على ذاكرة Word.	DIV_DI EN ENO ????-IN1 OUT-???? ????-IN2
١١	INC_B عملية إضافة تصاعدية بحجم Byte.	عمليات الإضافة تصاعدياً بحجم Byte بحيث يضاف واحد على الدخل IN و ينقل إلى الخرج OUT على ذاكرة بحجم Byte.	INC_B EN ENO ????-IN OUT-????
١٢	INC_W عملية إضافة تصاعدية بحجم Word.	عمليات ألإضافة تصاعدياً بحجم Word بحيث يضاف واحد على الدخل IN و ينقل إلى الخرج OUT على ذاكرة بحجم Word.	INC_W EN ENO ????-IN OUT-????
١٣	INC_DW عملية إضافة تصاعدية بحجم Dword.	عمليات ألإضافة تصاعدياً بحجم Dword بحيث يضاف واحد على الدخل IN و ينقل إلى الخرج OUT على ذاكرة بحجم Dword.	INC_DW EN ENO ????-IN OUT-????

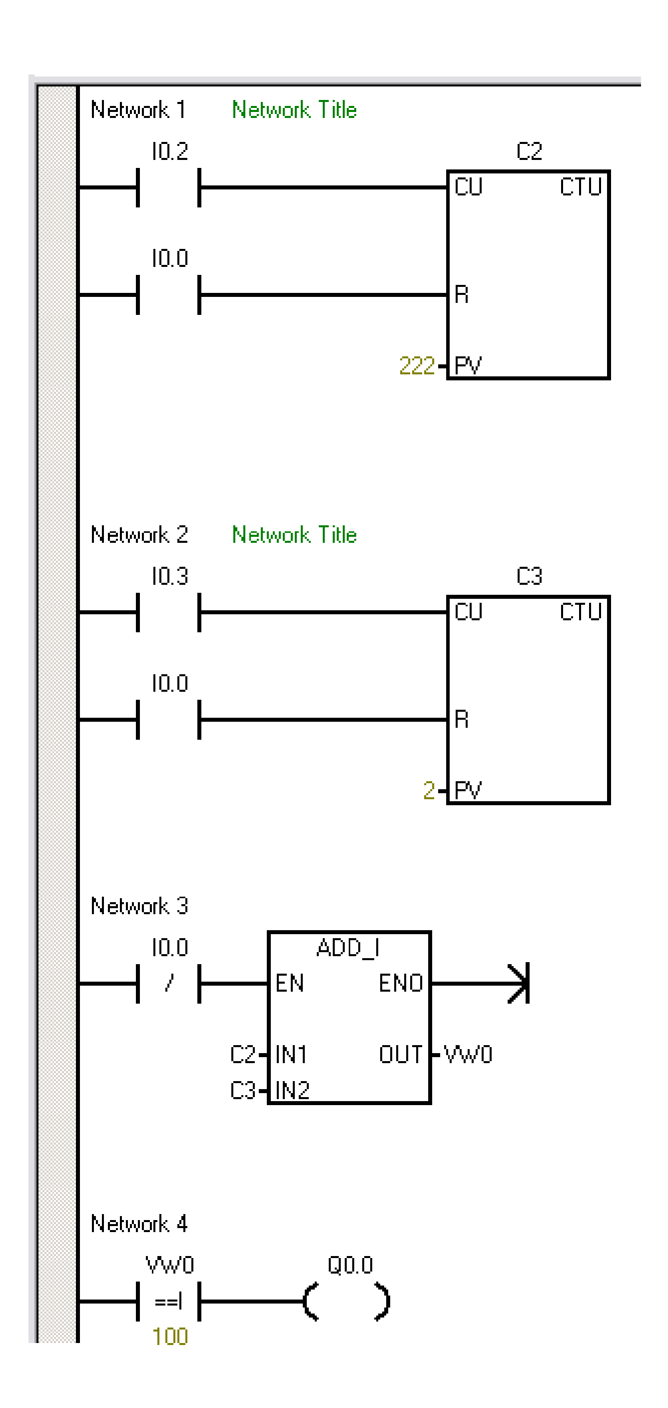
١٤	DEC_B عملية طرح تنازلية بحجم Byte.	عمليات الطرح تنازلياً بحجم Byte بحيث يطرح واحد من الدخل IN و ينقل إلى الخرج OUT على ذاكرة بحجم Byte.	DEC_B EN ENO ????-IN OUT-????
١٥	DEC_W عملية طرح تنازلية بحجم Word.	عمليات الطرح تنازلياً بحجم Word بحيث يطرح واحد من الدخل IN و ينقل إلى الخرج OUT على ذاكرة بحجم Word.	DEC_W EN ENO ????-IN OUT-????
١٦	DEC_DW عملية طرح تنازلية بحجم Dword.	عمليات الطرح تنازلياً بحجم Dword بحيث يطرح واحد من الدخل IN و ينقل إلى الخرج OUT على ذاكرة بحجم Dword.	DEC_DW EN ENO ????-IN OUT-????

أمثلة (تمارين عملية):

١– قم بتنفيذ دائرة تحكم منطقية لمصنع يحتوى على خطين أنتاج بحيث تضاء اللمبة عندما يكون حاصل مجموع القطع المنتجة من الخطين هو ١٠٠ قطعة.

عدد الدخل	نوع الدخل	أسم الدخل
١	n.o.	I0.0/S1
٢	n.o.	I0.2/S2
٣	n.o.	I0.3/S3
عدد العدادات	**نوع العدادات**	**أسم العدادات**
١	CTU	C2
٢	CTU	C3
عدد مفاتيح المقارنة	**نوع مفاتيح المقارنة**	**أسم مفاتيح المقارنة**
١	==I	VW0
عدد العمليات	**نوع العمليات**	**أسم العمليات**
١	ADD_I	ADD_I
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	لمبة	Q0.0

البرنامج:


Network 1 Network Title
I0.2
C2
CU CTU
I0.0
R
222
PV
Network 2 Network Title
I0.3
C3
CU CTU
I0.0
R
2
PV
Network 3
I0.0
ADD_I
EN ENO
C2 IN1 OUT VW0
C3 IN2
Network 4
VW0
==I
100
Q0.0

الشرح:

:Network1

عند مرور أى قطعة على خط الإنتاج الأول أمام الحساس I0.2 فأنة يرسل أشارة إلى العداد C2 بشرط أن يكون المفتاح I0.0 مفتوح.

:Network2

عند مرور أى قطعة على خط الإنتاج الثانى أمام الحساس I0.3 فأنة يرسل أشارة إلى العداد C3 بشرط أن يكون المفتاح I0.0 مفتوح.

:Network3

يقوم بجمع عدد القطع التى تم عدها سواء بواسطة العداد الخاص بخط الإنتاج الأول C2 أو بواسطة العداد الخاص بخط الإنتاج الثانى C3 وكتابة المجموع فى الذاكرة VW0.

:Network4

عندما تصبح قيمة الذاكرة VW0 تساوى ١٠٠ يصبح مفتاح المقارنة مغلق فتضاء اللمبة.

٢- قم بتنفيذ دائرة تحكم منطقية لمصنع يحتوى على خط أنتاج بحيث تمر الكرتونة أمام الحساس مع مراعاة أن كل كرتونة تحتوى على ١٢ زجاجة, صمم برنامج لمعرفة عدد الزجاجات و ليس الكراتين وتضاء لمبة أشارة عندما يصل عدد الزجاجات إلى ١٢٠.

عدد الدخل	نوع الدخل	أسم الدخل
١	n.o.	I0.0/S1
٢	n.o.	I0.1/S2
عدد العدادات	**نوع العدادات**	**أسم العدادات**
١	CTU	C0
عدد مفاتيح المقارنة	**نوع مفاتيح المقارنة**	**أسم مفاتيح المقارنة**
١	==I	VW24
عدد العمليات	**نوع العمليات**	**أسم العمليات**
١	MUL_I	MUL_I
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	لمبة	Q1.1/K1M

الشرح:

:Network1

عند مرور أى كرتونة على خط الإنتاج أمام الحساس I0.1 فأنة يرسل أشارة إلى العداد C0 بشرط أن يكون المفتاح I0.0 مفتوح.

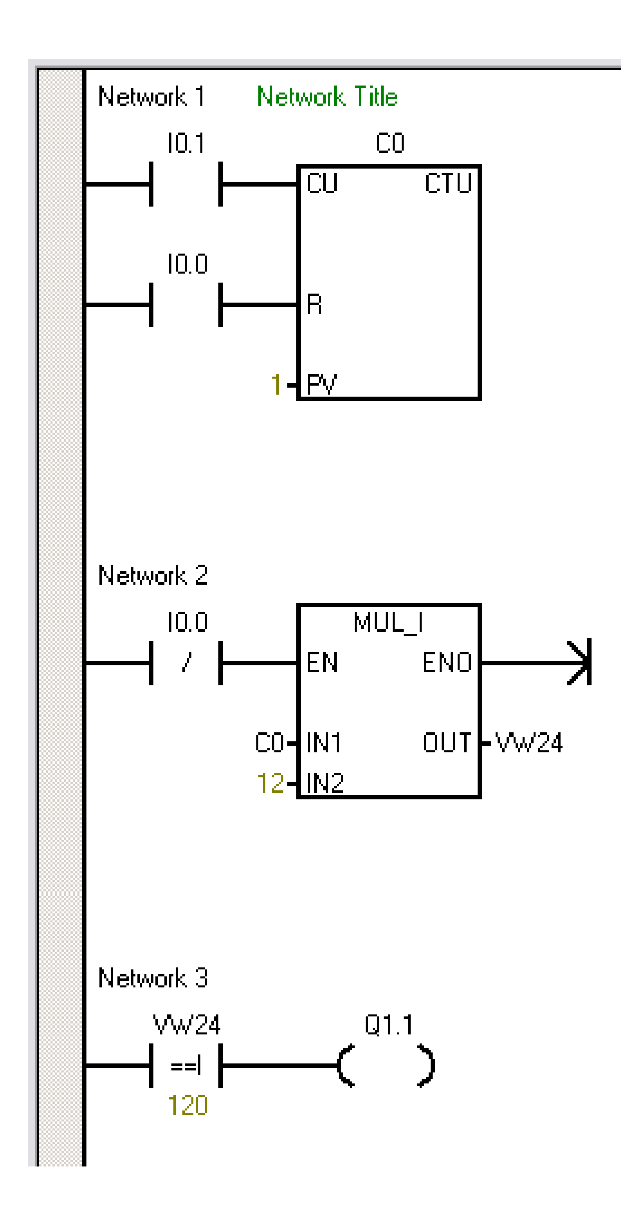
:Network2

يقوم البرنامج بضرب عدد الكراتين التى تم عدها بواسطة العداد C0 فى عدد الزجاجات ثم كتابة المجموع فى الذاكرة VW24.

:Network3

عندما تصبح قيمة الذاكرة VW24 تساوى ١٢٠ يصبح مفتاح المقارنة مغلق فتضاء اللمبة.

البرنامج:


Network 1
Network Title
I0.1
C0
CU
CTU
I0.0
R
1
PV
Network 2
I0.0
MUL_I
EN
ENO
C0
IN1
OUT
VW24
12
IN2
Network 3
VW24
==I
120
Q1.1

٣- قم بتنفيذ دائرة تحكم منطقية لمعادلة تقوم بتحويل درجة الحرارة من Kelvin إلى Celsius بحيث أذا كانت قيمة درجة الحرارة تحت الصفر تضيء لمبة حمراء.

علماً بأن المعادلة الخاصة بالتحويل هى. Kelvin=Celsius + 273

عدد الدخل	نوع الدخل	أسم الدخل
١	n.c.	I0.0/Stop
٢	n.o.	I0.1/Start
عدد المتغيرات	**نوع المتغيرات**	**أسم المتغيرات**
١	word	VW0(celsius)
٢	word	VW2(273)
٣	word	VW4(kelvin)
عدد العمليات	**نوع العمليات**	**أسم العمليات**
١	ADD_I	ADD_I
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	لمبة	Q1.1/K1M

الشرح:

:Network1

بالضغط على I0.1 وبشرط أن يكون I0.0 مغلق فيعمل الريليه M0.7

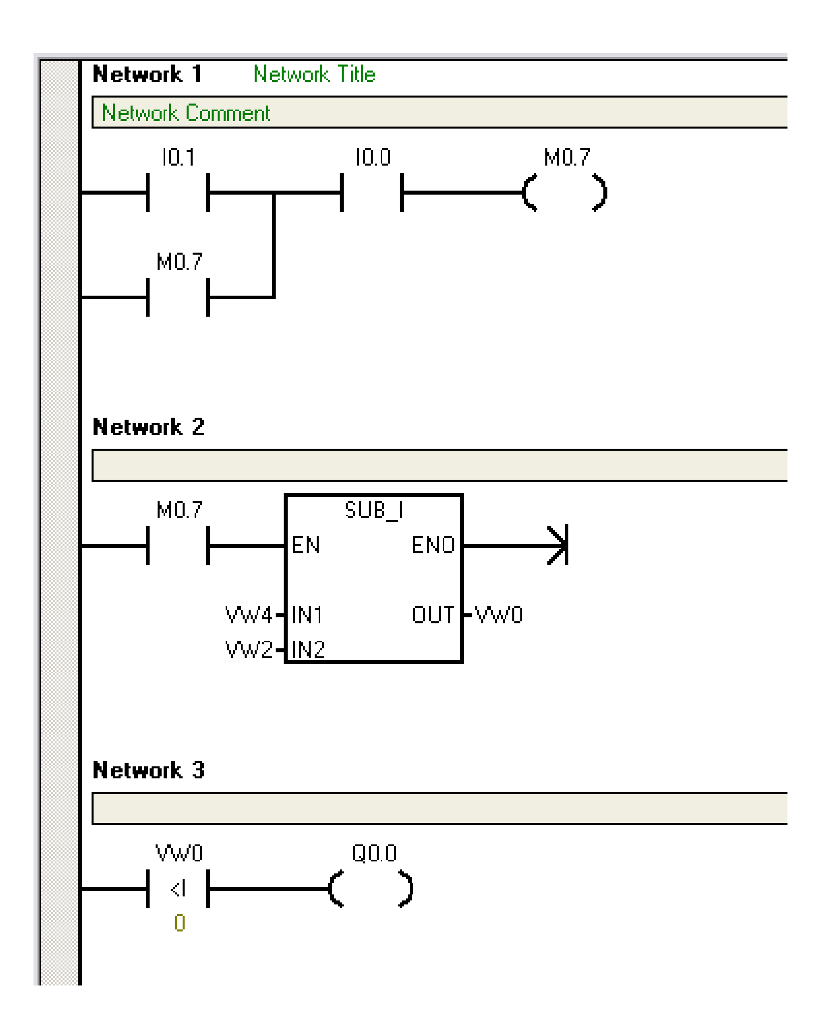
:Network2

يقوم بطرح قيمة المتغير VW2 الذى يمثل درجة الحرارة بال Kelvin من قيمة المتغير الآخرة VW2 التى تمثل الفرق بين القيمتين "273".

:Network3

عندما تصبح قيمة ال VW0 أقل من صفر سوف تضاء لمبة لتشير أن درجة الحرارة بالسالب.

البرنامج:


Network 1
Network Title
Network Comment
I0.1
I0.0
M0.7
M0.7
Network 2
M0.7
SUB_I
EN
ENO
VW4
IN1
OUT
VW0
VW2
IN2
Network 3
VW0
<I
0
Q0.0

٤- قم بتنفيذ دائرة تحكم منطقية لمعادلة تقوم بنفس عمل العداد حيث يعمل الخرج عندما يصل العدد إلى ٣٠٠٠٠٠ عدة.

علماً بأن المعادلة الخاصة بالعداد. VD0 = VD0 + 1

عدد الدخل	نوع الدخل	أسم الدخل
١	n.o.	I0.0/Stop
٢	n.o.	I0.1/Start
عدد المتغيرات	**نوع المتغيرات**	**أسم المتغيرات**
١	D.word	VD0
عدد العمليات	**نوع العمليات**	**أسم العمليات**
١	INC_DW	INC_DW
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	لمبة	Q1.0/lamp

الشرح:

:Network1

بالضغط على I0.1 سوف يتم إضافة واحد إلى المتغير VD0 فيعمل تماماً مثل العداد وقد تم استخدام مفتاح الـ positive edge لكى تكون كل ضغطة على المفتاح تعادل عدة واحدة فقط وليس أكثر.

:Network2

عندما تصبح قيمة المتغير VD0 تساوى 300000 فسوف يعمل الخرج Q1.0 تلقائياً.

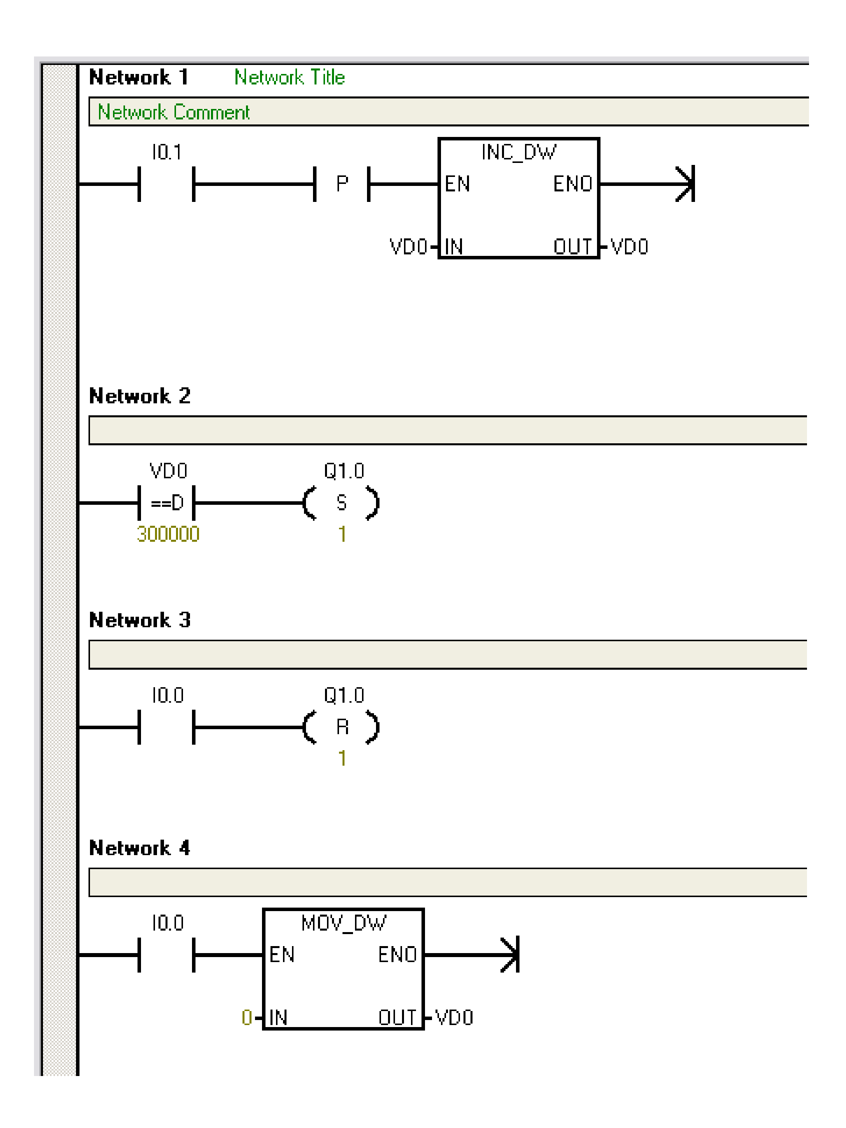
:Network3

بالضغط على I0.0 سوف يتم عمل reset أى إيقاف للخرج Q1.0.

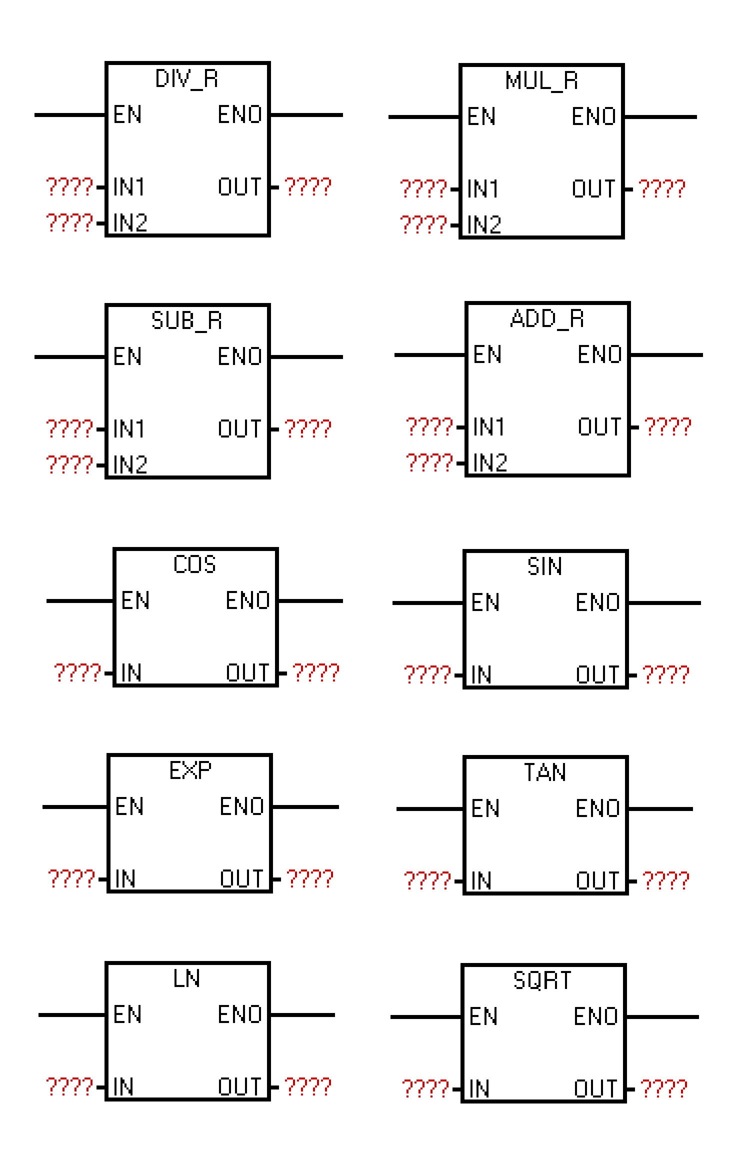
:Network4

بالضغط على I0.0 أيضاً سوف يتم نقل قيمة صفر إلى المتغير VD0 لكى يتمكن العامل من البدء من جديد بدايتاً من الصفر كما كان الوضع فى البداية.

البرنامج.


Network 1
Network Title
Network Comment
I0.1
P
INC_DW
EN
ENO
VD0
IN
OUT
VD0
Network 2
VD0
==D
300000
Q1.0
S
1
Network 3
I0.0
Q1.0
R
1
Network 4
I0.0
MOV_DW
EN
ENO
0
IN
OUT
VD0

العمليات الحسابية للأرقام العشرية:



كل العمليات التى تتم بواسطة العمليات الحسابية للأرقام العشرية يجب أن تحتوى فقط على أرقام عشرية و يجب أن تكون النواتج هى أيضاً عشرية فمثلاً:

- **لا يمكن استخدام العمليات الحسابية العشرية لجمع أرقام غير عشرية فمثلاً:**

  لا يمكن جمع رقم بقيمة ٢ و قيمة ٦ لأنها ليست أرقام عشرية.

- **لا يمكن استخدام العمليات الحسابية العشرية لطرح أرقام غير عشرية فمثلاً:**

  لا يمكن طرح رقم بقيمة ٨ من رقم ١ لأنها ليست أرقام عشرية.

- **لا يمكن استخدام العمليات الحسابية العشرية لقسمة أرقام غير عشرية فمثلاً:**

  لا يمكن قسمة رقم بقيمة ١ و قيمة ٣ لأنها ليست أرقام عشرية.

- **لا يمكن استخدام العمليات الحسابية العشرية لضرب أرقام غير عشرية فمثلاً:**

  لا يمكن ضرب رقم بقيمة ١٠٨ و قيمة ٢٣ لأنها ليست أرقام عشرية.

ملاحظة:

لا توجد مشكلة بالنسبة للمعادلة أى لا داعى لمراعاة الناتج من الناحية الخاصة بالذاكرة لأنه:

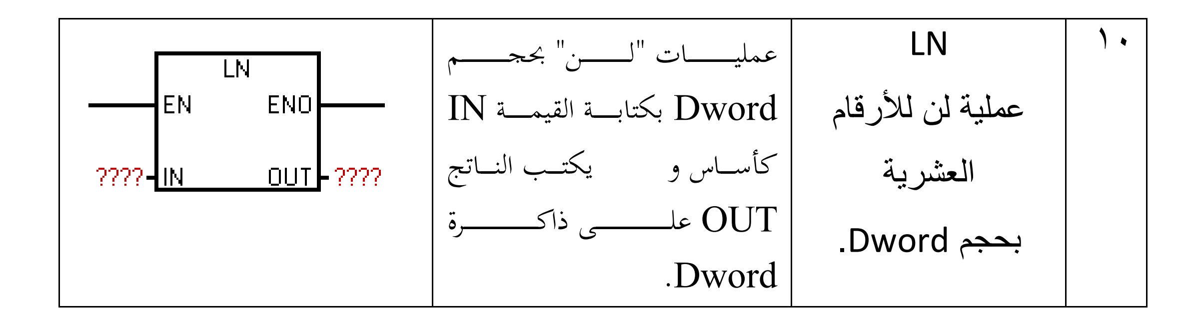
- يتم استخدام ذاكرة بحجم Dword مع العمليات الحسابية العشرية وهو أكبر حجم للذاكرة فى الـ PLC كما وضح فى الجزء الأول من هذا الكتاب.

لا توجد مشكلة بالنسبة للمعادلة أى لا داعى لمراعاة الناتج من الناحية الخاصة بنوع الناتج لأنه:

- فى حالة أن كان الناتج هو رقم صحيح بالصدفة مثلاً فلا توجد أى مشكلة لأنه يتم إضافة "٫٠" إلى الرقم فلا تتغير القيمة ولكن يصبح الرقم عشرى تلقائياً.

م	الاسم	الشرح	الشكل
١	ADD_R عملية جمع أرقام عشرية بحجم Dword.	عمليات جمع بحجم Dword تقوم بجمع أرقام عشرية (IN1 و IN2) ويكون الناتج (OUT) على ذاكرة Dword.	ADD_R EN ENO ????-IN1 OUT-???? ????-IN2
٢	SUB_R عملية طرح أرقام عشرية بحجم Dword.	عمليات طرح بحجم Dword تقوم بطرح أرقام عشرية (IN1 و IN2) ويكون الناتج (OUT) على ذاكرة Dword.	SUB_R EN ENO ????-IN1 OUT-???? ????-IN2
٣	MUL_R عملية ضرب أرقام عشرية بحجم Dword.	عمليات ضرب بحجم Dword تقوم بضرب أرقام عشرية (IN1 و IN2) ويكون الناتج (OUT) على ذاكرة Dword.	MUL_R EN ENO ????-IN1 OUT-???? ????-IN2
٤	DIV_R عملية ضرب أرقام عشرية بحجم Dword.	عمليات قسمة بحجم Dword تقوم بقسمة أرقام عشرية (IN1 و IN2) ويكون الناتج (OUT) على ذاكرة Dword.	DIV_R EN ENO ????-IN1 OUT-???? ????-IN2

٥	SQRT عملية جزر تربيعي أرقام عشرية بحجم Dword.	عمليات الجزر التربيعي بحجم Dword للدخل IN ويكتب الناتج OUT على ذاكرة Dword.	SQRT EN ENO ???? IN OUT ????
٦	COS عملية جتا للأرقام العشرية بحجم Dword.	عمليات "جتا" بحجم Dword للدخل IN ويكتب الناتج OUT على ذاكرة Dword.	COS EN ENO ???? IN OUT ????
٧	SIN عملية جا للأرقام العشرية بحجم Dword.	عمليات "جا" بحجم Dword للدخل IN ويكتب الناتج OUT على ذاكرة Dword.	SIN EN ENO ???? IN OUT ????
٨	TAN عملية ظا للأرقام العشرية بحجم Dword.	عمليات "ظا" بحجم Dword للدخل IN ويكتب الناتج OUT على ذاكرة Dword.	TAN EN ENO ???? IN OUT ????
٩	EXP عملية الأس للأرقام العشرية بحجم Dword.	عمليات "الأس" باستخدام قيمة ١٠ كأساس بحيث يكتب رقم الأس في الدخل IN ويكتب الناتج على OUT على ذاكرة Dword.	EXP EN ENO ???? IN OUT ????



١٠	LN عملية لن للأرقام العشرية بحجم Dword.	عمليات "لن" بحجم Dword بكتابة القيمة IN كأساس و يكتب الناتج على OUT ذاكرة Dword.	LN EN ENO ????- IN OUT -????

أمثلة (تمارين عملية):

١ – قم بتنفيذ دائرة تحكم منطقية لقانون فيثاغورث

علماً بأن المعادلة الخاصة بقانون فيثاغورث هي

$$C = \sqrt{a^2 + b^2}$$

عدد الدخل	نوع الدخل	أسم الدخل
١	n.c.	I0.0/Stop
٢	n.o.	I0.1/Start
عدد المتغيرات	**نوع المتغيرات**	**أسم المتغيرات**
١	D.word	VD0(a)
٢	D.word	VD2(b)
٣	D.word	VD4(C)
٤	D.word	VD20(a^2)
٥	D.word	VD22(b^2)
٦	D.word	VD40($\sqrt{a^2 + b^2}$)
عدد العمليات	**نوع العمليات**	**أسم العمليات**
١	ADD_R	ADD_R
٢	MUL_R	MUL_R
٣	MUL_R	MUL_R
٤	SQRT	SQRT

توضيح:

Network	العملية	المتغير	العملية	المتغير	أسم العملية
١	a x a	VD0 x VD0	a^2	VD20	عملية ضرب
٢	b x b	VD2 x VD2	b^2	VD22	عملية ضرب
٣	$a^2 + b^2$	VD20 + VD22	$a^2 + b^2$	VD40	عملية جمع
٤	$\sqrt{a^2 + b^2}$	SQRT(VD40)	$\sqrt{a^2 + b^2}$	VD4	جزر تربيعى

الشرح:

:Network1

بالضغط على I0.1 سوف يتم ضرب قيمة المتغير VD0 فى قيمة المتغير VD0 ويتم وضع الناتج فى المتغير .VD20

:Network2

بالضغط على I0.1 سوف يتم ضرب قيمة المتغير VD2 فى قيمة المتغير VD2 ويتم وضع الناتج فى المتغير .VD22

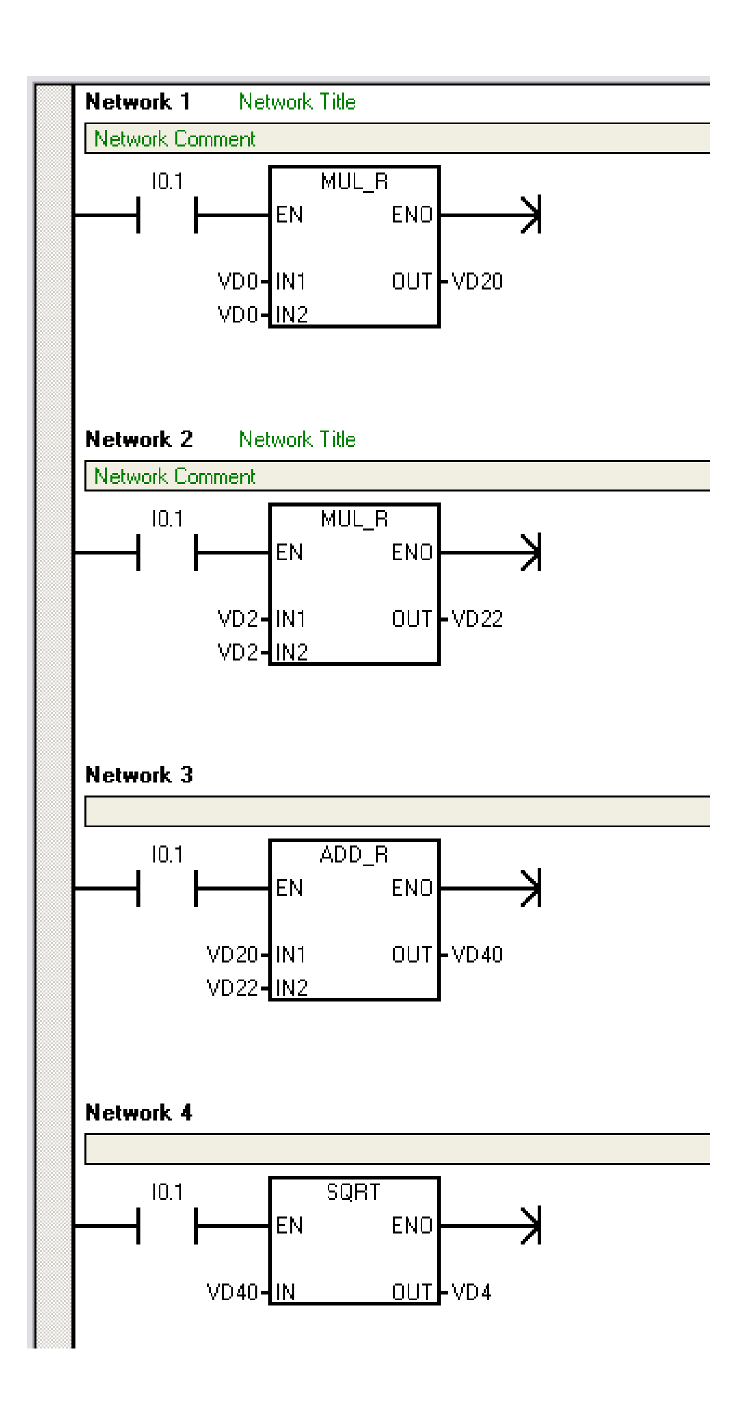
:Network3

بالضغط على I0.1 سوف يتم جمع قيمة المتغير VD20 وقيمة المتغير VD22 ويتم وضع الناتج فى المتغير .VD40

:Network4

بالضغط على I0.1 سوف يتم تطبيق الجذر التربيعى على قيمة المتغير VD40 ويتم وضع الناتج فى المتغير .VD4

البرنامج.


Network 1
Network Title
Network Comment
I0.1
MUL_R
EN
ENO
VD0
IN1
OUT
VD20
VD0
IN2
Network 2
Network Title
Network Comment
I0.1
MUL_R
EN
ENO
VD2
IN1
OUT
VD22
VD2
IN2
Network 3
I0.1
ADD_R
EN
ENO
VD20
IN1
OUT
VD40
VD22
IN2
Network 4
I0.1
SQRT
EN
ENO
VD40
IN
OUT
VD4

٢- قم بتنفيذ دائرة تحكم منطقية لمعادلة تقوم بتحويل الزاوية من Radiant إلى Degree.

علماً بأن المعادلة الخاصة بالتحويل هى.

$$D = \frac{G}{1.8} - 32$$

عدد الدخل	نوع الدخل	أسم الدخل
١	n.c.	I0.0/Stop
٢	n.o.	I0.1/Start
عدد المتغيرات	**نوع المتغيرات**	**أسم المتغيرات**
١	D.word	VD0(D)
٢	D.word	VD4(G)
٣	D.word	VD8(G/1.8)
عدد العمليات	**نوع العمليات**	**أسم العمليات**
١	ADD_R	ADD_R
٢	MUL_R	MUL_R
٣	MUL_R	MUL_R
٤	SQRT	SQRT

توضيح:

Network	العملية	المتغير	العملية	المتغير	أسم العملية
١	G/1.8	VD4 / 1.8	G/1.8	VD8	عملية قسمة
٢	(G/1.8) – 32	VD8 – 32	(G/1.8) – 32	VD0	عملية طرح

الشرح:

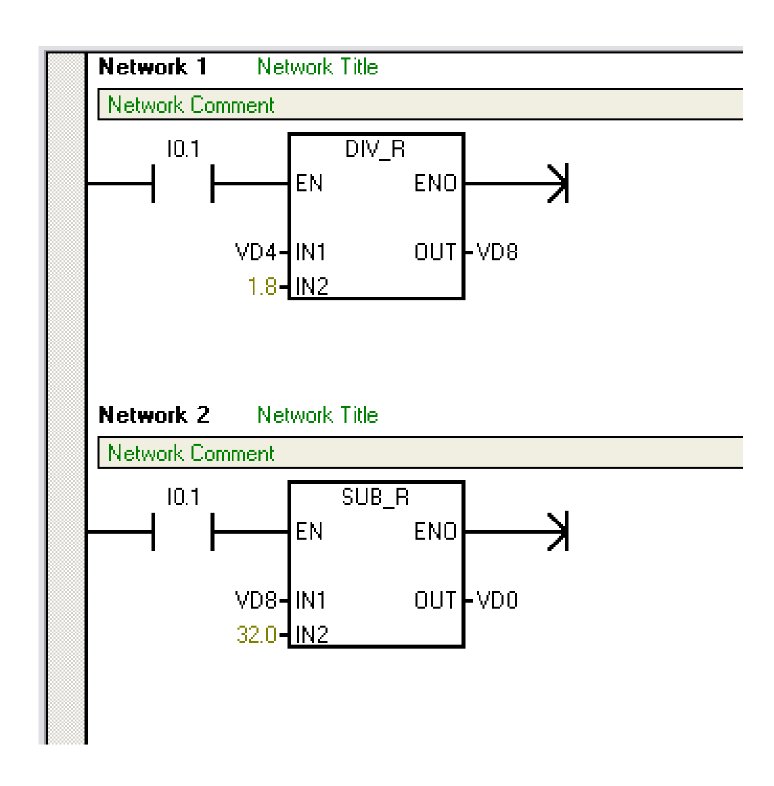
:Network1

بالضغط على I0.1 سوف يتم قسمة قيمة المتغير VD4 على ٨,١ ويتم وضع الناتج فى المتغير VD8.

:Network2

بالضغط على I0.1 سوف يتم طرح رقم ٣٢,٠ من قيمة المتغير VD8 ويتم وضع الناتج فى المتغير VD0.

البرنامج:



٣- قم بتنفيذ دائرة تحكم منطقية لمعادلة التالية:

$$Y = 2.5\,X^2 + \frac{3}{5}\,X - 4$$

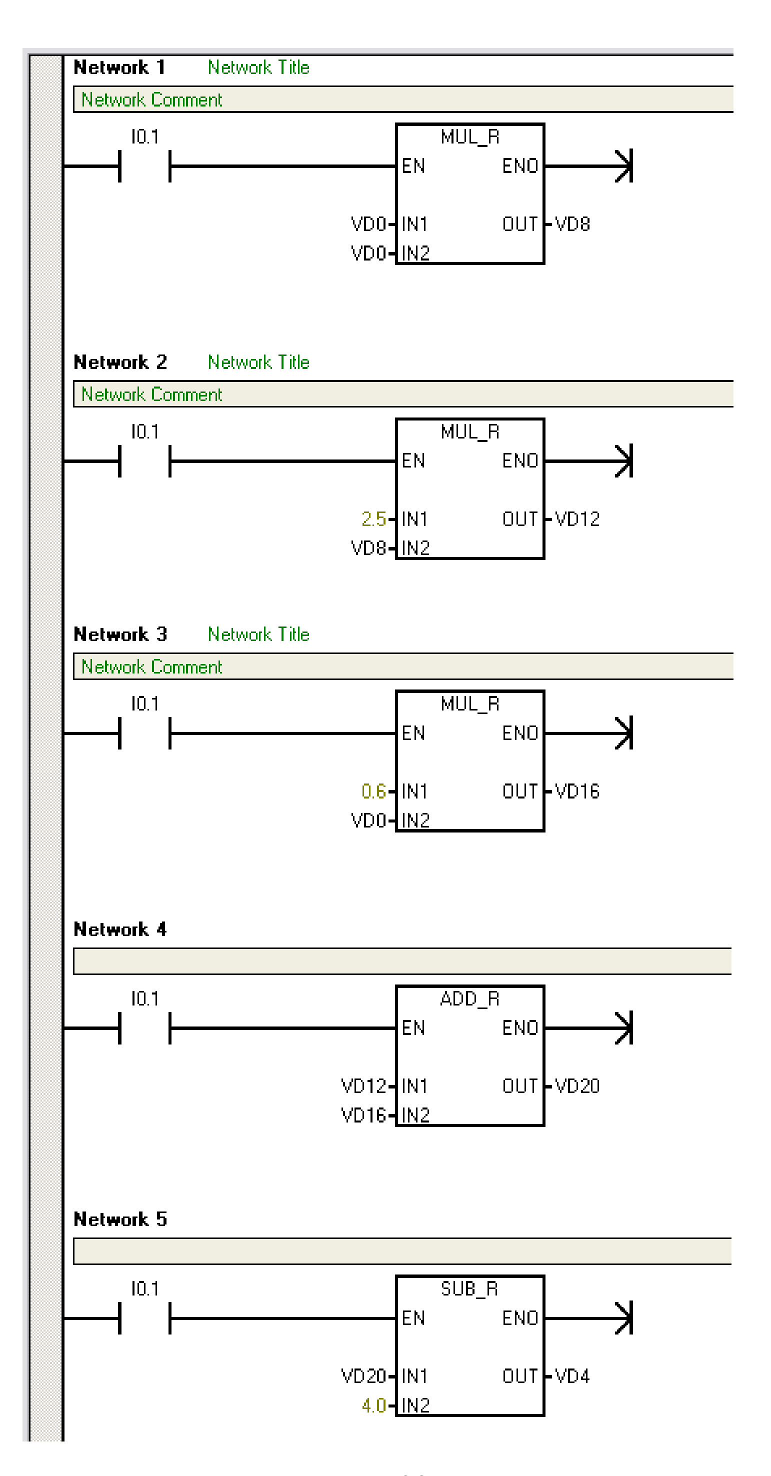
عدد الدخل	نوع الدخل	أسم الدخل
١	n.o.	I0.1/Start
عدد المتغيرات	**نوع المتغيرات**	**أسم المتغيرات**
١	word	VD0(x)
٢	word	VD4(Y)
عدد العمليات	**نوع العمليات**	**أسم العمليات**
١	ADD_R	ADD_R
٢	MUL_R	MUL_R
٣	MUL_R	MUL_R
٤	SQRT	SQRT

توضيح:

Network	العملية	المتغير	العملية	المتغير
١	x . x	VD0 x VD0	x^2	VD8
٢	$2,5 . x^2$	2.5 x VD8	$2.5x^2$	VD12
٣	0,6 . x	0.6 x VD0	0.6x	VD16
٤	$2,5 . x^2 + 0,6 . x$	VD12 + VD16	$2.5x^2+0.6x$	VD20
٥	$2,5 . x^2 + 0,6 . x - 4$	VD20 – 4.0	$2.5x^2+0.6x-4$	VD4

Network1	عملية ضرب	**Network4**	عملية جمع
Network2	عملية ضرب	**Network5**	عملية طرح
Network3	عملية ضرب		

البرنامج:


Network 1 Network Title
Network Comment
I0.1
MUL_R
EN ENO
VD0 IN1 OUT VD8
VD0 IN2
Network 2 Network Title
Network Comment
I0.1
MUL_R
EN ENO
2.5 IN1 OUT VD12
VD8 IN2
Network 3 Network Title
Network Comment
I0.1
MUL_R
EN ENO
0.6 IN1 OUT VD16
VD0 IN2
Network 4
I0.1
ADD_R
EN ENO
VD12 IN1 OUT VD20
VD16 IN2
Network 5
I0.1
SUB_R
EN ENO
VD20 IN1 OUT VD4
4.0 IN2

الشرح:

Network1:

بالضغط على I0.1 سوف يتم ضرب قيمة المتغير VD0 فى قيمة المتغير VD0 ويتم وضع الناتج فى المتغير VD8.

Network2:

بالضغط على I0.1 سوف يتم ضرب رقم ٥,٢ فى قيمة المتغير VD8 ويتم وضع الناتج فى المتغير VD12.

Network3:

بالضغط على I0.1 سوف يتم ضرب رقم ٦,٠ فى قيمة المتغير VD0 ويتم وضع الناتج فى المتغير VD16.

Network4:

بالضغط على I0.1 سوف يتم جمع قيمة المتغير VD12 فى قيمة المتغير VD16 ويتم وضع الناتج فى المتغير VD20.

Network5:

بالضغط على I0.1 سوف يتم طرح رقم ٤ من قيمة المتغير VD20 ويتم وضع الناتج فى المتغير VD4.

ملاحظة:

لقد تم تنفيذ المعادلة السابقة كمثال عابر ولكن فى حقيقة الأمر قد تكون هذه المعادلة خاصة بقيمة تناظرية تشير إلى درجة حرارة, سرعة محرك, شدة الضغط, إلخ....

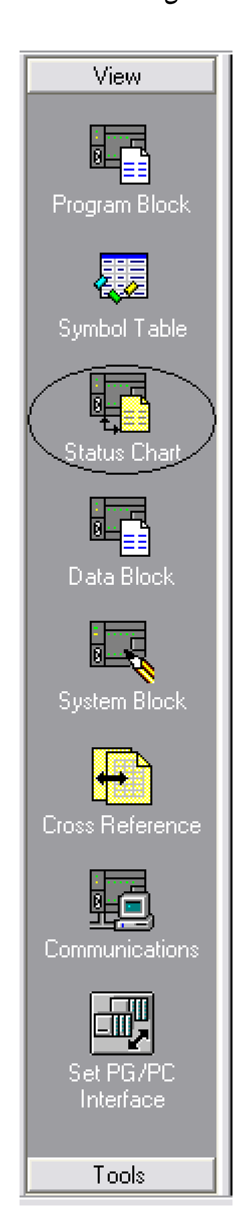
الباب الثانى

جدول الحالات

- محتويات جدول الحـــــــــــــالات.
- أستخدامات جدول الحـــــــــــــالات.
- المفاتيح المستخدمة فى جدول الحالات.
- الطرق المستخدمة فى أظهار حالة العناوين.
- كيفية كتابة مجموعة عناويـــــــــــــن.
- أمـــــــــــــــــــــــــــر Write all.
- أمـــــــــــــــــــــــــــر Force.
- كيفية تطبيق أمر Write all على تمارين.
- كيفية تطبيق أمر Force على تمـــــارين.
- الرسم التخطيطى لأى عنوان فى البرنامـــج.

جدول الحالات Status Chart:

تستخدم صفحة "جدول الحالات" الـ Status Chart لمعرفة الحالة الخاصة بكل عنوان من العناوين المستخدمة أو غير المستخدمة في البرنامج و يمكن أيضاً تعديل حالة المدخلات, المخرجات, الريليهات, المؤقتات الزمنية, العدادات و المتغيرات بواسطة هذه الصفحة.



طريقه استخدام صفحة "جدول الحالات ":

- بواسطة صفحة "جدول الحالات" الـ Status Chart يمكن معرفة حالة المدخلات و المخرجات وأي عنوان أخر دون الذهاب إلى مكان المداخل أو مكان المخرجات, فمثلاً قد يوجد من ضن مجموعة المفاتيح (حساس موصل أسفل مكينة معينة) فبدلاً من الذهاب للبحث عن المفتاح لمعرفة إذا كان مفتوح أو مغلق فيمكن معرفة الحالة بواسطة صفحة "جدول الحالات" بكل سهولة.

تستخدم أيضاً صفحة "جدول الحالات" الـ Status Chart لمعرفة الأعطال:

- فمثلا فى حالة أن كان المفتاح مغلق بينما تقول صفحة "جدول الحالات" أنة مفتوح فهذا يعنى أن المفتاح به مشكلة (ليس موصل بالكهرباء – تم التوصيل على العزل – الكهرباء الموصلة بالمفتاح أقل من الحد المسموح بة – مشكلة بنقاط التلامس الخاصة بالمفتاح – إلخ..).
- بينما فى حاله إذا كان الخرج (المحرك) لا يعمل بينما تشير صفحة جدول الحالات أن الخرج يعمل فهذا يشير إلى عطل غير مسئول عنه وحدة البرمجة نفسها (قد لا توجد تغذية لدائرة القوة – مشكلة بالريليه الميكانيكى أو بنقطة الريلية الذى يعمل كحماية بين وحدة البرمجة ودائرة القوى – نقاط القاطع الحرارى مفتوحة).

ملاحظة. سوف يتم شرح الأعطال بالتفصيل بجميع أنواعها فى الكتاب التالى.

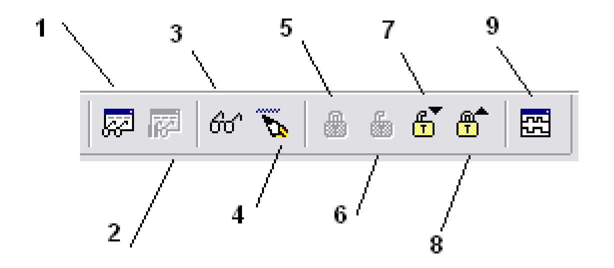
الشكل العام لصفحة **"جدول الحالات"**:

	Address	Format	Current Value	New Value
1	I0.0	Bit		
2	Q0.3	Bit		
3	T32	Bit		
4	C11	Bit		
5	T32	Signed		
6	C11	Signed		

المقصود بكلمة (Address – Format – Current value – New value)

- Address: حيث تكتب العناوين وليس بالشرط أن يكون العنوان مستخدم فى البرنامج.
- Format: حيث يتم أختيار الطريقة المراد أظهار بها حالة العنوان.
- Current value: حيث تظهر حالة العنوان.
- New value: حيث يمكن تعديل حالة العنوان.

مفاتيح هامة فى صفحة "جدول الحالات":



١- Chart status: لمعرفة حالة العناوين المكتوبة بصفة مستمرة.

٢- Pause chart status: لإيقاف عند نقطة معينة لمعرفة حالة العناوين المكتوبة.

٣- single read: لمعرفة حالة العناوين المكتوبة فى لحظة معينة.

٤- Write all: لتعديل أو تغير بعض العناوين ولكن حسب البرنامج اى حسب الشروط المستخدمة فى البرنامج, فمثلاً لا يمكن تشغيل محرك عكس حركة فى الاتجاهين معاً وهذا ليس ذكاء من جهاز الـ PLC لأنه لا يعرف مقدماً أنه سوف يحدث قفله بل لأن شروط البرنامج تشير إلى أنه لا يمكن للاتجاهين أن يعمل معاً فى وقت واحد.

٥- **Force:** لتعديل أو تغير بعض العناوين مهما كان البرنامج اى أنه لا تؤخذ فى الاعتبار الشروط المستخدمة, فمثلاً يمكن تشغيل محرك عكس حركة فى الاتجاهين معاً لذالك يجب وضع الحمايات ليس فقط فى البرنامج بل على الريلية الميكانيكى إيضاً كم وضح فى الجزء الأول من هذا الكتاب.

٦- Unforce: لإلغاء أمر Force بالنسبة لـ bit اى لدخل واحد أو لخرج واحد فقط.

٧- Unforce all: لإلغاء أمر Force بالنسبة لـ byte أو word أو Dword أى لمجموعة مدخلات أو لمجموعة مخرجات.

٨- Read all forced: لمعرفة كل العناوين التى طبق عليها أمر Force خاصاً عندما تتعامل مع الجهاز لأول مرة يجب التأكد من أن كل المخارج تعمل بطريقة طبيعية وليس تحت تأثير أمر force.

٩- Trend view: بالضغط عليها يقوم جهاز الـ PLC بعمل رسم تخطيطى diagram للعناوين المكتوبة بصفحة "جدول الحالات" سواء كانت العناوين الموجودة هى (مدخلات – مخرجات – مؤقتات زمنية – عدادات – متغيرات – ريليهات داخلية – إلخ.....).

طريقة أظهار حاله العنوان الـ format :

Format: هى الطريقة المراد أظهار بها حالة العنوان و تنقسم إلى:

- Unsigned - Signed - Hexadecimal - Binary - Bit -
- ASCII - Floating point

الشكل العام لكل format:

- **Bit**: 2#0 حيث رقم اثنان يشير أن اللغة المستخدمة هى Binary.
- **Binary**: 2#0000_0000 حيث رقم "اثنان" يشير أن اللغة المستخدمة هى Binary.
- **Hexadecimal**: 16#2A حيث رقم "ستة عشر" يشير أن اللغة المستخدمة هى Hexa.
- **Signed**: ٧٨٥٨+ أو ٥٩٣- حيث يحتوى الرقم على أشارة سواء موجبة أو سالبة.
- **Unsigned**: ٦٧١١ حيث لا يحتوى الرقم على أشارة موجبة أو سالبة.
- **Floating point**: ٢ E+ ٥٣٩, ٢+ حيث أن الرقم المكتوب هو رقم عشرى.
- **ASCII**: "S" أو "<" حيث يشير إلى الحالة باستخدام حروف أو رموز.

الشرح:

- بعض العناوين لا يمكن أن تظهر بأكثر من **format**, فمثلاً المفتاح I0.1 أو الخرج Q2.6 أو الريليه M1.1 أو المتغيرات V23.4 لا يمكن أن يكون لهم **format** غير ال **Bit** وهذا لأن حالة الدخل أو الخرج أو الريليه أو المتغير أما أن تكون واحد أو صفر.

- يستخدم ال **format** (Floating point) مع الأرقام العشرية, ومع الأرقام العشرية ومع المعدلات أو المتغيرات أو المداخل والمخارج التناظرية.

- يعتبر ال **format** (ASCII) من ال **format** الأقل استخداماً حيث يظهر الحالة الخاصة بالعنوان المذكور باستخدام أحرف أو رموز بدلاً من استخدام الأرقام.

- بعض العناوين يمكن أن تعمل مع أكثر من **format**, فمثلاً المؤقت الزمنى T32 أو العداد C0 أو المتغير VB4 يمكن أن يكون لهم **format** مختلفة مثل ال Binary أو Signed أو Unsigned أو Hexadecimal أو ASCII ولكن يفضل دائماً اختيار الأفضل حسب نوع العنوان المستخدم.

- بالنسبة للمؤقت الزمنى أو للعداد يمكن تكرار العنوان مرتين بحيث أن تكون المرة الأولى مثلاً format (Bit) وهى تعنى الحالة الخاصة بنقطة العداد أو نقطة المؤقت الزمنى بحيث تشير أذا كانت مغلقة أو مفتوحة بينما المرة الثانية تكون مثلاً FORMAT (Signed, Unsigned, Binary أو Hexadecimal) بحيث تشير إلى قيمة الرقمية للعداد أو القيمة الرقمية للمؤقت الزمنى.

- فى حالة الاستعلام عن حالة مجموعة من المدخلات أو المخرجات فبدلاً من كتابة كل دخل أو كل خرج أو كل ريليه أو كل متغير على حدى يمكن الاستعلام عن مجموعة مكونة من ٨ أو ١٦ أو ٣٢ دخل أو خرج أو ريليه أو متغير معاً عن طريق حرف ال B الذى يرمز إلى ال **Byte** أى ثمانية

أو عن طريق حرف الـ W الذى يرمز إلى الـ Word أى ستة عشر أو عن طريق حرف الـ D الذى يرمز إلى الـ Dword أى اثنان و ثلاثون فمثلاً:

شرح مبسط

- IB0: تعنى أول "ثمانية مدخلات".
- IB1: تعنى ثانى "ثمانية مدخلات".
- QB6: تعنى سابع "ثمانية مخرجات".
- QB9: تعنى عاشر "ثمانية مخرجات ".
- IW2: تعنى ثانى "ستة عشر دخل".
- IW6: تعنى رابع "ستة عشر دخل ".
- QW4: تعنى ثلاث "ستة عشر خرج ".
- QW8: تعنى خامس "ستة عشر خرج ".
- ID0: تعنى أول " اثنان وثلاثين دخل ".
- ID4: تعنى ثانى " أثنان وثلاثين دخل ".
- QD8: تعنى ثلاث "اثنان وثلاثين مخرج".
- QD12: تعنى رابع "اثنان وثلاثين مخرج".

للتوضيح أنظر صفحة ٢٠١ فى الجزء الأول من هذا الكتاب.

باستخدام صفحة "جدول الحالات" الـ Status Chart التى تستخدم لمعرفة الحالة الخاصة بكل عنوان من العناوين المستخدمة فى البرنامج أو غير المستخدمة حتى, يمكن أيضاً تعديل أو تغيير حاله العناوين المستخدمة فى البرنامج وغير المستخدمة أيضاً فلذالك توجد طريقتان للتعديل أو التغير فى البرنامج:

الطريقة الأولى هى write all. 

الطريقة الثانية هى force: 

الطريقة الأولى للتعديل فى البرنامج.

تستخدم Write all فى تعديل البرنامج ولكن "**مع مراعاة الشروط**" المستخدمة فى البرنامج.
المقصود بكلمة **مراعاة الشروط** المستخدمة فى البرنامج أى أنه لا يمكن تنفيذ أى أمر بواسطة الـ write all وفى نفس الوقت يكون من غير الممكن تنفيذ هذا الأمر بواسطة البرنامج, فمثلاً:

أ. لا يمكن لمحرك أن يعمل بينما يكون مفتاح الإيقاف الخاص بالمحرك مفتوح.

ب. لا يمكن لمحرك أن يقف بينما يكون مفتاح التشغيل الخاص بالمحرك مغلق.

ت. لا يمكن لمحرك عكس حركة (يعمل فى اتجاهين) أن يعمل فى الاتجاهين معاً فى نفس الوقت.

ث. لا يمكن لمؤقت زمنى أن يبدأ بالعمل بينما لا تكون هناك تغذية.

ج. لا يمكن لعداد أن يبدأ بالعد بينما تكون كل النقاط الخاصة بالإشارة غير مغلقة.

فلذالك ولا حتى أمر write all يستطيع أن يقوم بتنفيذ أى من النقاط السابقة

الطريقة:

١ – تتم كتابة العناوين المراد التعامل معها سواء كانت, مخرجات أو مؤقتات زمنية أو عدادات أو متغيرات إلخ.....

	Address
1	I0.1
2	Q0.4
3	T32
4	C14
5	VW0

٢– يتم اختيار الطريقة format المراد إظهار بها حالة العنوان المذكور أعلاه.

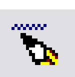
Format
Bit
Binary
Signed
Unsigned
Hexadecimal
ASCII

٣– يتم الضغط على مفتاح chart status لمعرفة حالة العناوين بصفة مستمرة.

Current Value
2#1
2#1
+11103
+2
+30000

٤ – يكتب فى new value مقابل كل عنوان التعديل المراد تنفيذه.

New Value
2#0
2#1
+1000
+30
+0

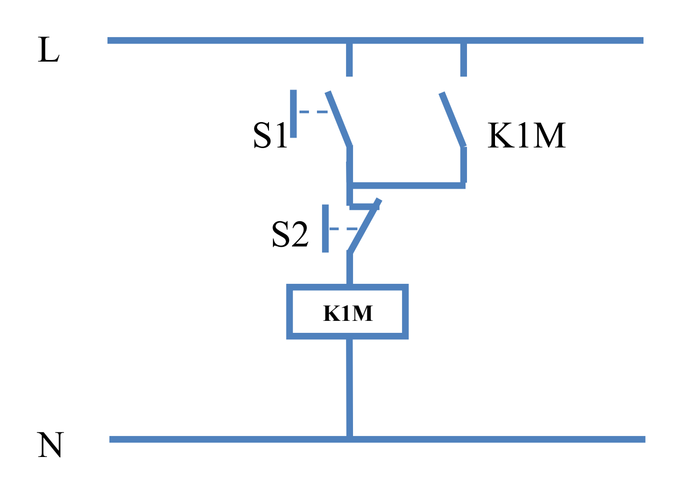
بعد ذالك يتم الضغط على write all  **لتنفيذ التغيير المراد تطبيقه**

ملاحظة:

- لا يمكن تعديل حالة المدخلات باستخدام أمر Write all.
- فى حالة تعديل خرج يكتب واحد للتشغيل أو صفر للإيقاف.
- فى حالة تعديل قيمة العداد تكتب القيمة المراد العداد الوصول إليها وليس بالضرورى أن يتم تنفيذ هذا التعديل أثناء عمل العداد.
- فى حالة تعديل قيمة المؤقت الزمنى تكتب القيمة المراد الوصول إليها بواسطة المؤقت الزمنى ولكن بشرط أن يكون تنفيذ هذا التعديل أثناء عمل المؤقت الزمنى.
- فى حالة كتابة أى قيمة على متغيرات يجب أن يأخذ فى الاعتبار بأن القيمة المراد كتابتها بواسطة أمر write all يجب أن لا تكون أكبر من أكبر رقم يمكن كتابته على المتغيرات وينطبق هذا على جميع المتغيرات بمختلف أحجامها حيث أن فى حاله كتابة قيم كبيرة على متغيرات صغيرة قد يسبب هذا مشاكل سوف يتم توضيحها فى الكتاب التالى الخاص بالأعطال والتمرين العملية.

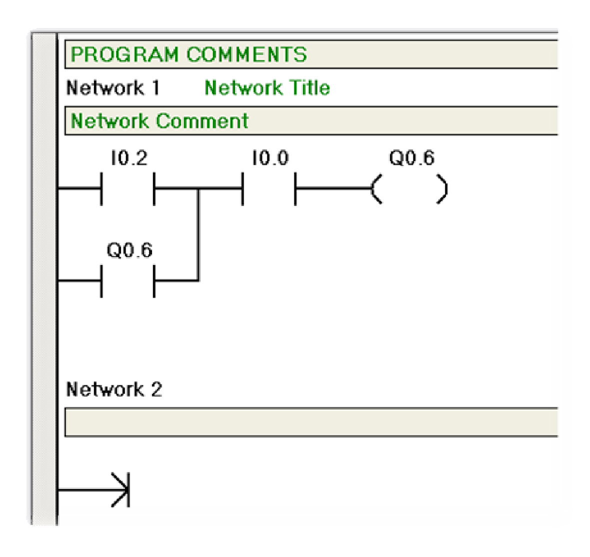
مثال باستخدام أمر Write all:

محرك يعمل من مكان واحد.



عدد الدخل	نوع الدخل	أسم الدخل
١	n.o.	I0.2/S1
٢	n.c.	I0.0/S2
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتكتور	Q0.6/K1M

البرنامج:



الخطوات:

- أولاً: يكتب العنوان المراد التعامل معه فى Address و فى هذا التمرين العنوان هو Q0.6
- ثانياً: يتم اختيار الـ Format حسب الـ Address و فى هذه التمرين الـ Format هو Bit فقط.
- ثالثاً: تكتب الحالة المراد تعديلها أى 2#0 للإيقاف و 2#1 للتشغيل.
- رابعاً: يتم الضغط على Write all لتنفيذ التعديل المرغوب فيه.

	Address	Format	Current Value	New Value
1	Q0.6	Bit		2#1

ملاحظة:

لإلغاء أمر Write all الذى تم استخدامه فى المثال السابق توجد طريقتان:

١– التحكم بالخرج عن طريق البرنامج أى بفتح I0.0 وهذا لأن أمر write all لا يستطيع أن يخالف البرنامج نفسه من حيث طريقة التحكم بالخرج.

٢– التحكم بالخرج عن طريق Status chart أى للإيقاف يكتب 2#0 و للتشغيل يكتب 2#1 ثم الضغط على Write all مرة أخرى.

	Address	Format	Current Value	New Value
1	Q0.6	Bit		2#0

يمكن أيضاً تغير قيم المؤقتات الزمنية بواسطة نفس الأمر:

الخطوات:

- **أولاً:** يكتب العنوان المراد التعامل معه فى Address و فى هذه الحالة العنوان هو T32.
- **ثانياً:** يتم اختيار الـ Format حسب الـ Address و فى هذه الحالة الـ Format هو Bit بالنسبة لنقطة المؤقت الزمنى و Signed بالنسبة للقيمة الفعلية للمؤقت الزمنى.
- **ثالثاً:** تكتب الحالة المراد تعديلها أى لتغيير القيمة إلى أربع ثوانى تكتب قيمة ٤٠٠٠.
- **رابعاً:** يتم الضغط على Write all لتنفيذ التعديل المرغوب فيه.

بشرط أن يتم هذا التطبيق أثناء عمل المؤقت الزمنى لأننا نتحدث عن زمن حقيقى فلذالك ليس من المنطقى أن يتم أعطاء أى زمن للمؤقت ثم يتم بدء حسب هذا مؤخراً عند تشغيل المؤقت الزمنى فهذا لأننا نتحدث عن زمن حقيقى real time.

	Address	Format	Current Value	New Value
1	T32	Signed		+4000

يمكن أيضاً تغير قيم العدادات:

الخطوات:

- **أولاً**: يكتب العنوان المراد التعامل معه فى Address و فى هذه الحالة العنوان هو C10.
- **ثانياً**: يتم اختيار الـ Format حسب الـ Address و فى هذه الحالة الـ Format هو Bit بالنسبة لنقطة المؤقت الزمنى و Signed بالنسبة للقيمة الفعلية للعداد.
- **ثالثاً**: تكتب الحالة المراد تعديلها أى لتغيير القيمة إلى خمسة عشر تكتب قيمة ١٥.
- **رابعاً**: يتم الضغط على Write all لتنفيذ التعديل المرغوب فيه.

	Address	Format	Current Value	New Value
1	C10	Signed		+15

ملاحظة:

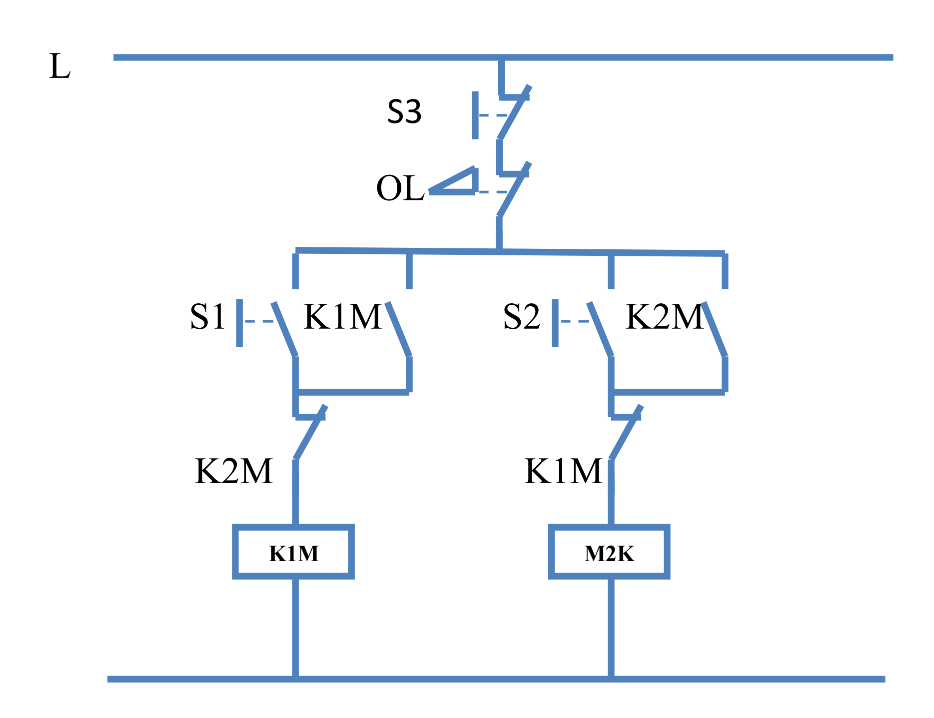
- يجب دائماً الضغط على Write all لتنفيذ القيم المكتوبة فى New value.

- ليس بالشرط أن يتم هذا التطبيق أثناء عمل العداد لأننا نتحدث عن إشارات حقيقية تحتسب فقط عند إرسال الإشارة بواسطة الضغط على المفتاح مثلاً فلذالك من المنطقى جداً أن يتم أعطاء أى رقم للعداد ثم يتم بدء حسب هذا مؤخراً عند تشغيل العداد أو فى الحال إذ كان العداد يعمل أساساً فهذه العملية لا ترتبط بزمن حقيقى real time.

- فى حالة إرسال قيمة إلى العداد حتى وأن كانت هذه القيمة أكبر أو تساوى القيمة المسبقة (إى القيمة المقصودة) فأنه يشترط دائما أن يعمل العداد حتى يتم تفعيل الحاله الحقيقية للنقاط الخاصة بالعداد لأنه ليس من المنطقي أن يغير العداد النقاط الخاصة به أن كان لم يعمل بعد.

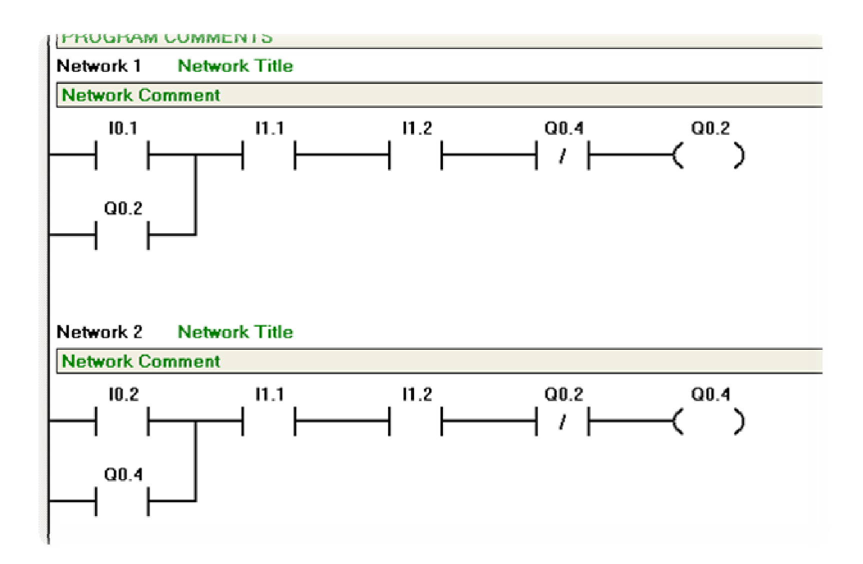
مثال أخر باستخدام أمر **Write all**:

محرك يعمل فى اتجاهين:



عدد الدخل	نوع الدخل	أسم الدخل
١	n.o.	I0.1 / S1
٢	n.o.	I0.2 / S2
٣	n.c.	I1.1 / S3
٤	n.c.	I1.2 / OL
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتكتور	Q0.2 / K1M
٢	كونتكتور	Q0.4 / K2M

البرنامج



الخطوات:

- **أولاً:** يكتب العنوان المراد التعامل معه فى Address و فى هذا التمرين العنوان هو Q0.2 و Q0.4
- **ثانياً:** يتم اختيار الـ Format حسب الـ Address و فى هذه التمرين الـ Format هو Bit فقط.
- **ثالثاً:** تكتب الحالة المراد تعديلها أى 2#0 للإيقاف أو 2#1 للتشغيل بالنسبة لأى من الخرجين.
- **رابعاً:** يتم الضغط على Write all لتنفيذ التعديل المرغوب فيه.

	Address	Format	Current Value	New Value
1	Q0.2	Bit		2#0
2	Q0.4	Bit		2#1

ملاحظة:

- لإلغاء أمر Write all الذى تم استخدامه فى المثال السابق تتم استخدام نفس الطريقتان الذين سبق شرحهم.
- من الشيء المهم جداً فى الـ Write all أنه لا يمكن أن يقوم بتغير أى حالة دون مراعاة الشروط الموجودة فى البرنامج, فمثلاً التمرين السابق هو عبارة عن محرك اتجاهين ومن المعروف أنة لا يمكن تشغيل اتجاهين معاً لأن التمرين يحتوى على نقط مغلقة من الطرفين فلذلك ولا حتى أمر Write all يستطيع أن يقوم بتشغيل الاتجاهين معاً.

الطريقة الثانية للتعديل فى البرنامج.

تستخدم Force فى تعديل البرنامج "**وبدون أى مراعاة للشروط**" المستخدمة فى البرنامج.
المقصود بكلمة **بدون مراعاة الشروط** المستخدمة فى البرنامج أى أنه يمكن تنفيذ أى أمر بواسطة الـ force وفى نفس الوقت يكون هذا الأمر غير ممكن تنفيذه بواسطة البرنامج فمثلاً:

أ. يمكن لمحرك أن يعمل بينما يكون مفتاح الإيقاف الخاص بالمحرك مفتوح.
ب. يمكن لمحرك أن يقف بينما يكون مفتاح التشغيل الخاص بالمحرك مغلق.
ت. يمكن لمحرك عكس حركة أن يعمل فى الاتجاهين معاً فى نفس الوقت (قفله).

الطريقة:

١- تتم كتابة العناوين المراد التعامل معها سواء كانت, مداخل, مخارج, الخ........

Address

٢- يتم اختيار الطريقة format المراد أظهار بها حالة العنوان المذكور أعلاه كما وضح فى ما قبل فى نفس الباب.

Format
Bit
Binary
Signed
Unsigned
Hexadecimal
ASCII

٣- يتم الضغط على chart status لمعرفة حالة العناوين المكتوبة بصفة مستمرة.

Current Value
2#1
2#1
+11103
+2
+30000

٤- يكتب مقابل العنوان new value التعديل المراد تنفيذه.

New Value
2#0
2#1
+1000
+30
+0

ملاحظة:

- يمكن تعديل حالة الدخل باستخدام أمر Force.
- فى حالة تعديل خرج يكتب واحد للتشغيل أو صفر للإيقاف.
- لا يمكن تعديل قيمة العداد أو المؤقت الزمنى باستخدام أمر Force.

- في حاله تغيير أى خرج إلى 2#1 لا يعني أن إلغاء الحالة سيكون عن طريق 2#0 لأن في الحالتين لا تراعى الأولويات أو الشروط الخاصة بالبرنامج.

بعد ذالك يتم الضغط على Force.

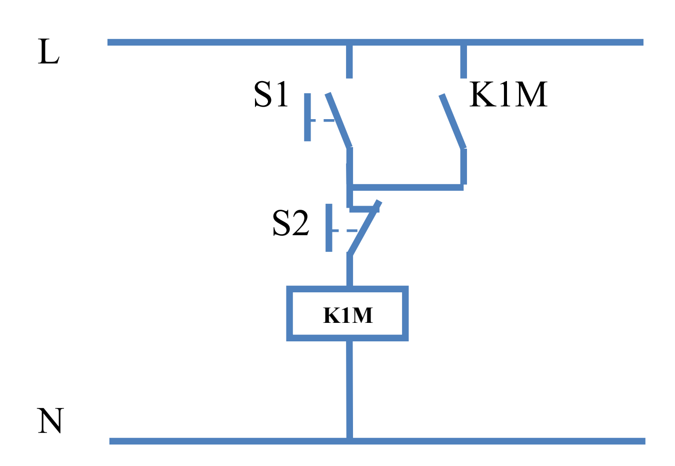
لإلغاء أمر Force بالنسبة لعنوان واحد فأنة يتم الضغط على Unforce.

لإلغاء أمر Force بالنسبة لكل العناوين فأنة يتم الضغط على Unforce all.

لمعرفة أى العناوين التى تم تطبيق أمر Force عليها فأنة يتم الضغط على Read all forced.

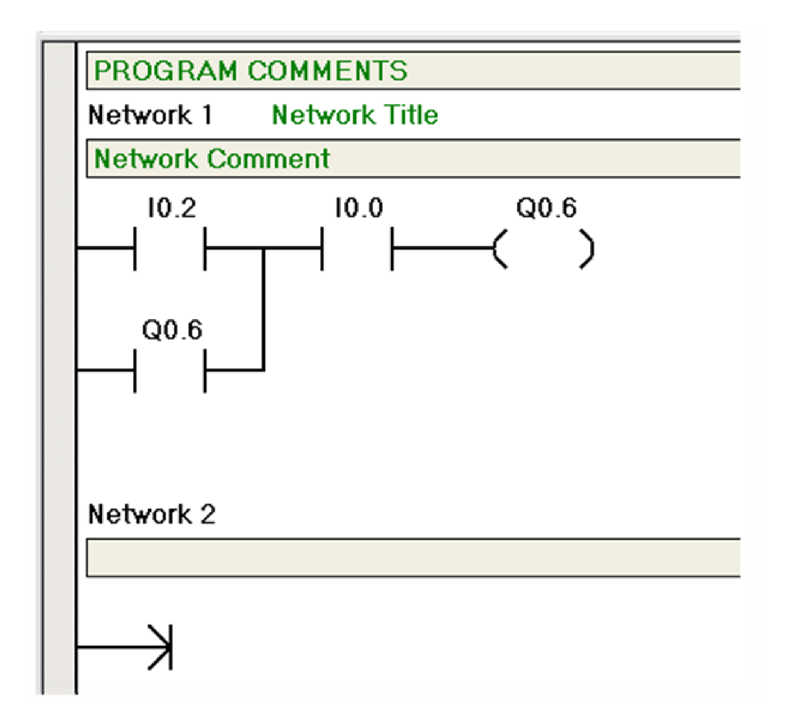
مثال باستخدام أمر Force:

محرك يعمل من مكان واحد.



عدد الدخل	نوع الدخل	أسم الدخل
١	n.o.	I0.2/S1
٢	n.c.	I0.0/S2
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتكتور	Q0.6/K1M

البرنامج



الخطوات:

- **أولاً**: يكتب العنوان المراد التعامل معه فى address و فى هذا التمرين العنوان هو Q0.6
- **ثانياً**: يتم اختيار الـ format حسب الـ address و فى هذه التمرين الـ format هو bit فقط.
- **ثالثاً**: تكتب الحالة المراد تعديلها أى 2#0 للإيقاف أو 2#1 للتشغيل.
- **رابعاً**: يتم الضغط على force لتنفيذ التعديل المرغوب فيه.

	Address	Format	Current Value	New Value
1	Q0.6	Bit		2#1

ملاحظة:

لإلغاء أمر Force الذى تم استخدامه فى المثال السابق توجد طريق واحدة فقط وهى:

- لإلغاء أمر Force يستخدم أمر Unforce دون كتابة أى شيء.
- أمر Force بقيمة 2#0 ليس عكسها Force بقيمة 2#1 و العكس صحيح أيضاً ولكن لإلغاء أمر Force دائماً وأبداً يستخدم أمر Unforce.

	Address	Format	Current Value	New Value
1	Q0.6	Bit		2#0

- لا يمكن تغير قيم المؤقتات الزمنية باستخدام force:

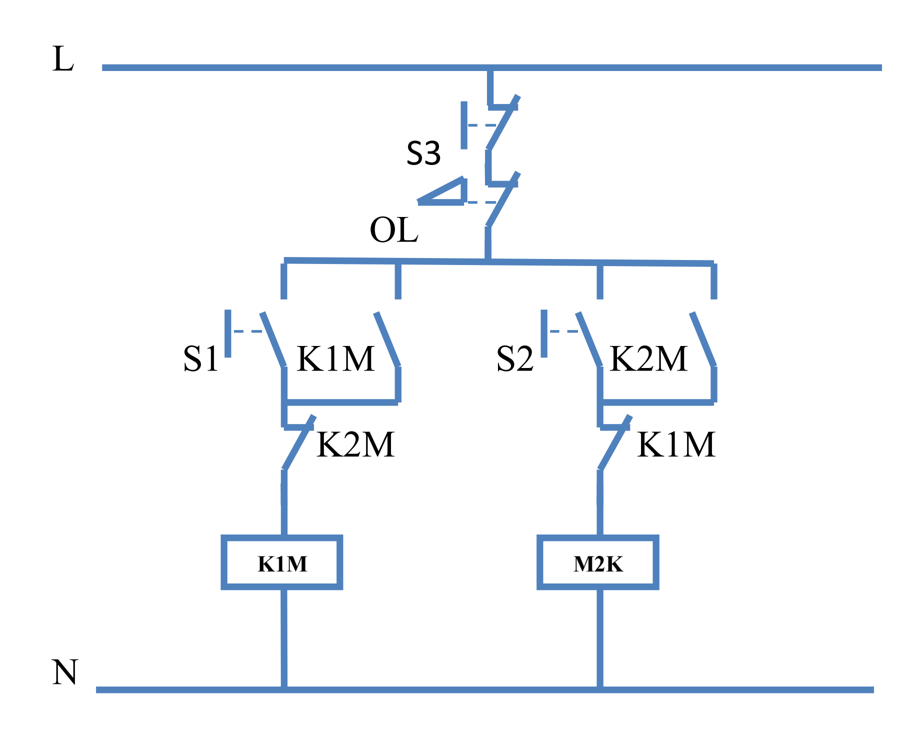
	Address	Format	Current Value	New Value
1	T32	Signed		+4000

- لا يمكن تغير قيم العدادات باستخدام force:

	Address	Format	Current Value	New Value
1	C10	Signed		+15

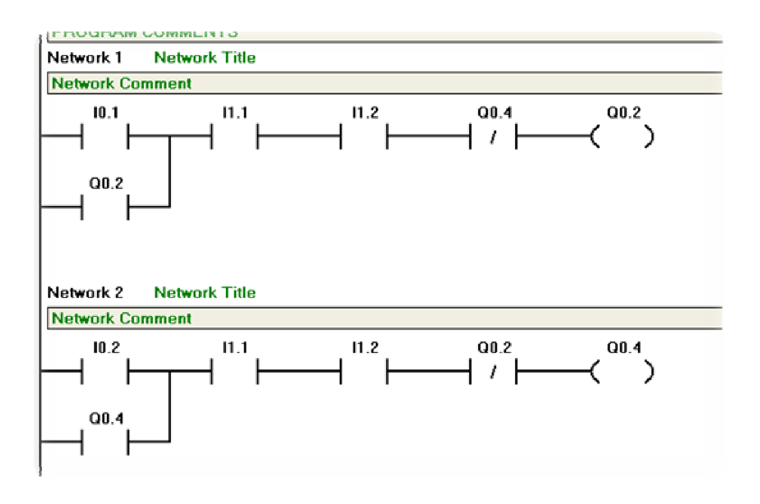
مثال أخر باستخدام أمر **Force**:

محرك يعمل فى اتجاهين:



عدد الدخل	نوع الدخل	أسم الدخل
١	n.o.	I0.1 / S1
٢	n.o.	I0.2 / S2
٣	n.c.	I1.1 / S3
٤	n.c.	I1.2 / OL
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتكتور	Q0.2 / K1M
٢	كونتكتور	Q0.4 / K2M

البرنامج:



الخطوات:

- **أولاً:** يكتب العنوان المراد التعامل معه فى address و فى هذا التمرين العنوان هو Q0.2 و Q0.4
- **ثانياً:** يتم اختيار الـ format حسب الـ address و فى هذا التمرين الـ format هو bit فقط.
- **ثالثاً:** تكتب الحالة المراد تعديلها أى 2#0 للإيقاف أو 2#1 للتشغيل بالنسبة للخرجين.
- **رابعاً:** يتم الضغط على force لتنفيذ التعديل المرغوب فيه.

	Address	Format	Current Value	New Value
1	Q0.2	Bit	2#0	2#1
2	Q0.4	Bit	2#1	2#0

ملاحظة:

- لإلغاء أمر Force الذى تم استخدامه فى المثال السابق تتم استخدام نفس الطريقة التى تم شرحها فى ما سبق.

- من الشيء المهم جداً فى ال force أنه يمكن أن يقوم بتغير أى حالة دون مراعاة الشروط الموجودة فى البرنامج, فمثلاً التمرين السابق هو عبارة عن محرك اتجاهين ومن المعروف أنة لا يمكن تشغيل اتجاهين معاً لأن التمرين يحتوى على نقط مغلقة من الطرفين ولكن يستطيع أمر force أن يقوم بتشغيل الاتجاهين معاً وهذا يعنى أنه ستحدث "قفلة" على أطراف المحرك.

- أمر force يستطيع تغير قيمة الدخل أيضاً.

- نظراً إلى أن أمر force من الأوامر التى أذا أنسيت قد تسبب المشاكل فأنة تظهر علامة قفل مغلق بالقرب من العنوان الذى تم تطبيق أمر force عليه.

- أذا تم تطبيق أمر force بقيمة 2#1 على الخرج Q0.0 فأنه سيعمل فقط هذا الخرج, أى أنة أذا كان البرنامج قد صمم بحيث عندما يعمل هذا الخرج يعمل أيضاً معه مؤقت زمنى فأنة لن سيعمل لأن الخرج لم يعمل بطريقة طبيعية بل عمل عن بواسطة أمر force.

يمكن استخدام أمر force فى حل بعض الأعطال, فمثلاً:

- أذا كان المحرك لا يعمل فيمكن بدايتاً تطبيق أمر Force على كل مفتاح من المفاتيح التى تتحكم بالمحرك على حدا بحيث أنه قد يكون هناك مشكلة فى دخل معين فإذا عمل المحرك عند تنفيذ أمر force على المفتاح I2.3 مثلاً فهذا يعنى أنه توجد مشكلة بتلك المفتاح.

أمر ال.رسم التخطيطى Trend view:

يعتبر هذا الأمر من العماليات التوضيحية الموجودة بالبرنامج حيث يقوم بعمل "رسم تخطيطى" لكل العناوين المذكورة فى صفحة Status chart.

الخطوات:

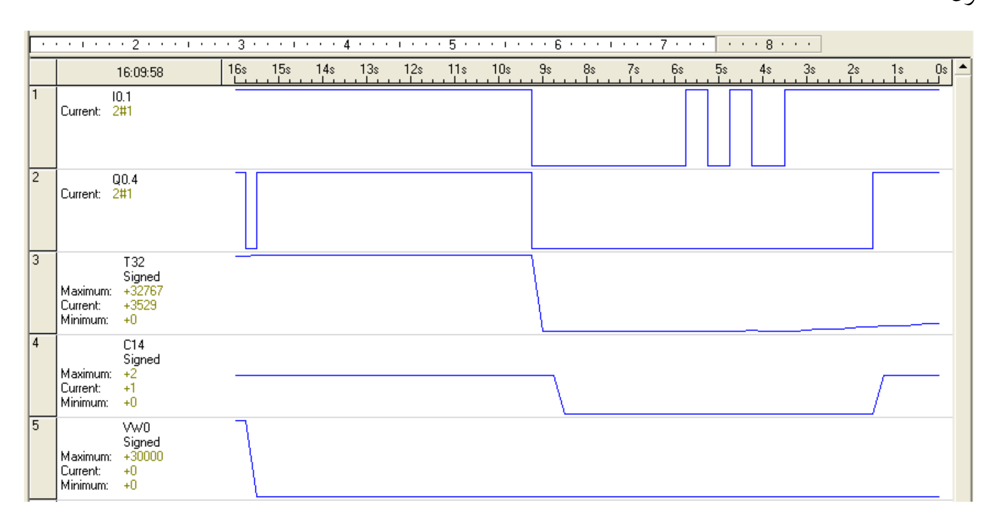
١– تكتب العناوين المراد إظهارها فى "الرسم التخطيطى" فى جدول الحالات.

	Address	Format	Current Value	New Value
1	I0.1	Bit		
2	Q0.4	Bit		
3	T32	Signed		
4	C14	Signed		
5	VW0	Signed		

٢– يتم الضغط على chart status لمعرفة القيمة الحالية لكل عنوان من الخمسة عناوين المذكورة فى الجدول.

	Address	Format	Current Value	New Value
1	I0.1	Bit	2#1	
2	Q0.4	Bit	2#1	
3	T32	Signed	+32767	
4	C14	Signed	+2	
5	VW0	Signed	+30000	

٣– يتم الضغط على trend view لرأيه الرسم التخطيطى لكل عنوان من الخمسة عناوين المذكورة فى الجدول.

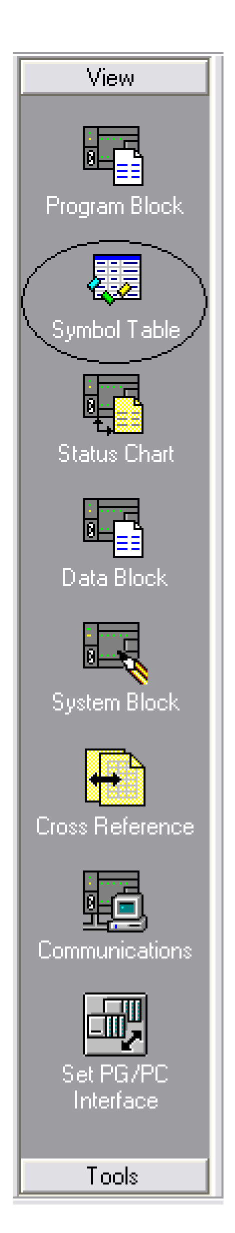


الباب الثالث

جدول الرموز

- شرح جـــــــــــــــــــدول الرموز.
- محتويات جـــــــــــــــــدول الرموز.
- المفاتيح المستخدمة فى صـــــفحة الرموز.
- الأخـــــــــــــطاء الممكن التعرض لها.
- التعليقات الخاصة بصفحة جـــدول الرموز.
- جدول الرموز الخاص بـــصفحات البرمجة.
- طريقة البرمـــــــــــجة باستخدام العناوين.
- طريقة البرمجة باستخدام الرمـــــــــوز.
- جـــــــــدول الـ S7-200 symbols.

جدول الرموز symbol table :



تستخدم صفحة "جدول الرموز" الـ Symbol Table لكى يكون لكل عنوان من العناوين المستخدمة فى البرنامج رمز و يمكن أيضاً كتابة تعليقات خاصة بكل رمز حتى يصبح من السهل على المبرمج التعرف على وظيفة كل مفتاح, ريليه, خرج, عداد, مؤقت زمنى, إلخ.

طريقة استخدام صفحة "جدول الرموز":

- يمكن عمل البرنامج أولاً ثم إعطاء رموز لكل عنوان بواسطة صفحة "جدول الرموز".
- يمكن كتابة الرموز أولاً فى صفحة "جدول الرموز" ثم عمل البرنامج حسب الرموز المعطاة أو المحددة من قبل.

الأخطاء الممكن حدوثها:

١-كتابة أى رمز يكون بالصدفة أسم من أسماء العمليات المتعارف عليها فى البرمجة.

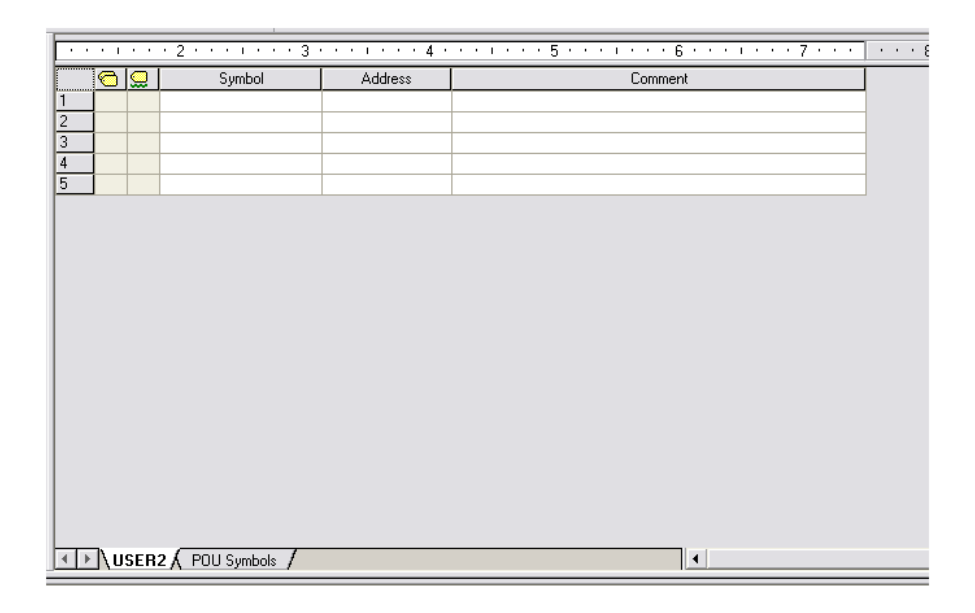
- أى أنه لا يمكن كتابة كلمة ("stop", "S", "R", "JMP", "LBL", "END", "ENI", "TON", "TOF", "TONR", "CTU", "CTD", "CTUD") لأن كل هذه الكلمات هى دوال لعمليات تستخدم فى البرمجة كما سيتم شرحها فيما بعد.

٢-كتابة أى رمز مكون من كلمتين و يكون بينهم مسافة.

- أى أنه لا يمكن كتابة كلمة ("first button", "second button ", "motor right", "motor left").

٣- تكرار نفس الرمز أكثر من مرة.

- أى أنه لا يمكن كتابة أى رمز أكثر من مرة حتى ولو كان مع نفس العنوان.



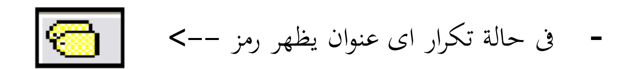
ملاحظة:

Symbol: حيث يكتب الرمز الخاص بكل عنوان بشرط عدم تكرار الرمز مع عناوين أخرى.

Address: حيث يكتب العنوان الخاص بالبرنامج بشرط أن يكون العنوان مستخدم فى البرنامج بالفعل.

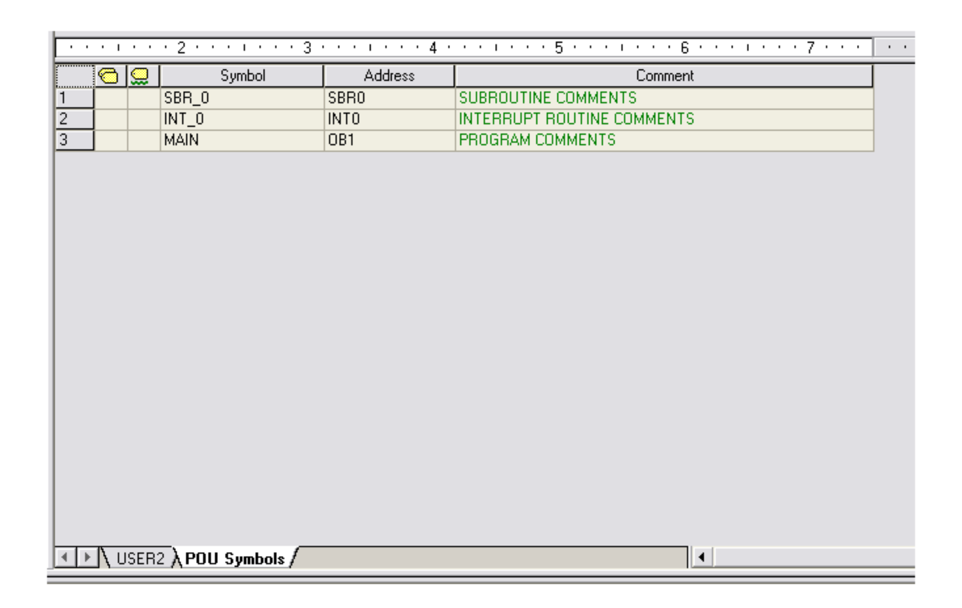
Comment: حيث تكتب الملاحظات الخاص بكل عنوان مستخدم فى البرنامج.

- فى حالة تكرار اى عنوان يظهر رمز -->



- فى حالة كتابة اى عنوان ليس مستخدم فى البرنامج يظهر رمز -->





ملاحظة:

Symbol: حيث توجد الرموز الخاصة بالثلاث أماكن الخاصة بصفحات البرمجة.

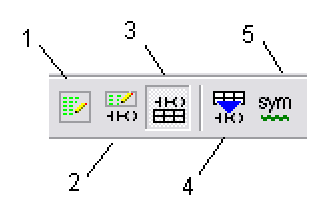
Address: حيث توجد العناوين الخاصة بالثلاث أماكن الخاصة بصفحات البرمجة.

Comment: حيث توجد الملاحظات الخاصة بالثلاث أماكن الخاصة بصفحات البرمجة.

ملاحظة:

- لا يمكن تعديل الرمز أو العنوان أو التعليقات الخاصة بصفحات البرمجة الثلاثة.
- فى حالة إضافة صفح أخرى خاصة بالبرمجة يضاف الرمز الخاص بها تلقائيا فى صفحة "جدول الرموز".

المفاتيح المستخدمة فى صفحة الرموز.



١- Toggles POU comments:

- بالضغط على هذا المفتاح تظهر التعليقات الخاصة بصفحات البرمجة.

PROGRAM COMMENTS

٢- Toggles NETWORK comments:

- بالضغط على هذا المفتاح تظهر التعليقات الخاصة بفروع البرمجة.

Network Comment

٣- Toggles symbol information table:

- بالضغط على هذا المفتاح تظهر الرموز و التعليقات الخاصة بكل عنوان فى جدول.

Symbol	Address	Comment

٤- Apply all symbols in project:

- فى حالة كتابة الرموز بعد الانتهاء من البرمجة يتم الضغط على هذا المفتاح لكى يقوم بإظهار الرموز بجوار العناوين.

٥- Create table undefined symbols:

- فى حاله عمل برنامج باستخدام الرموز ودون كتابة أى عناوين يتم الضغط على هذا المفتاح لكى يقوم البرنامج تلقائيا بإظهار الرموز التى ليس لها عناوين فى جدول لكي يقوم المبرمج بكتابة العناوين.

توجد طريقتان لاستخدام جدول الرموز فى البرمجة:

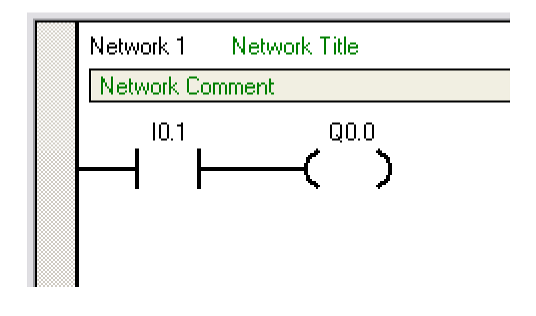
١ – البرمجة باستخدام العناوين ثم إعطاء رمز لكل عنوان.

٢ – البرمجة باستخدام الرموز ثم إعطاء عنوان لكل رمز.

الطريقة الأولى:

البرمجة باستخدام العناوين أولاً ثم إعطاء رمز لكل عنوان.

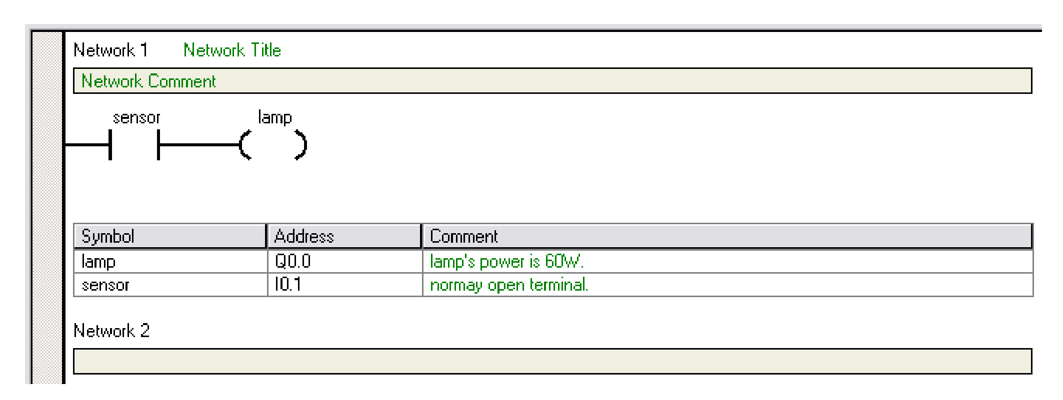
١ - رسم البرنامج باستخدام العناوين.



٢ - كتابة رمز لكل عنوان.

			Symbol	Address	Comment
1			sensor	I0.1	normay open terminal.
2			lamp	Q0.0	lamp's power is 60W.
3					
4					
5					

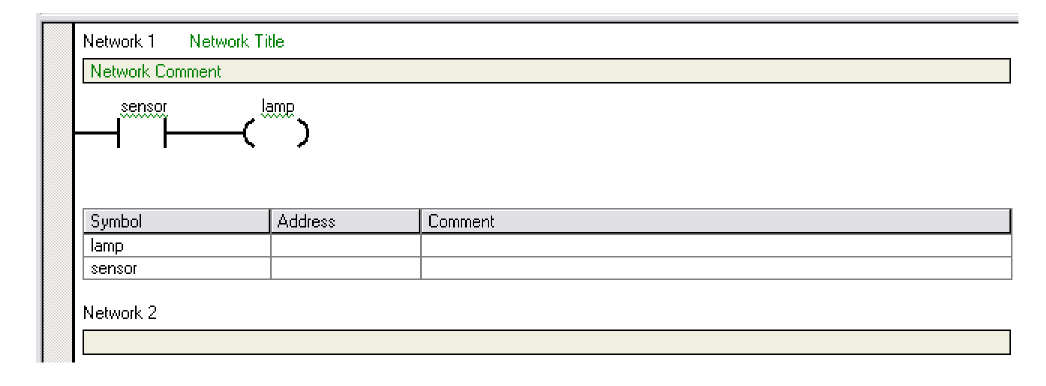
٣ -الضغط على Apply all symbols in project لتفعيل التغيير.



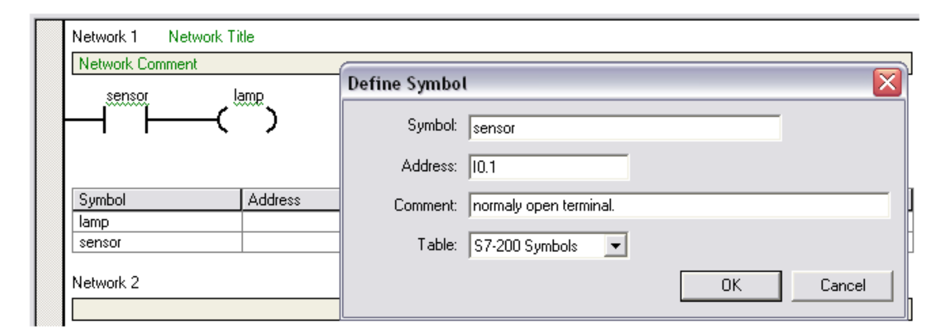
الطريقة الثانية:

البرمجة باستخدام الرموز أولاً ثم إعطاء عنوان لكل رمز.

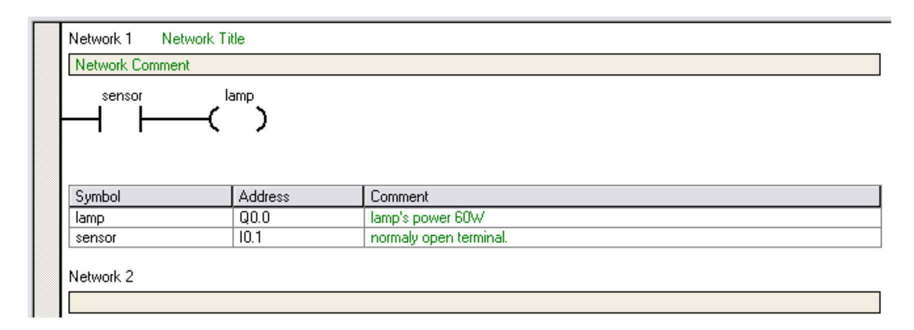
١ - رسم البرنامج باستخدام الرموز.



٢ - كتابة عنوان لكل رمز بالضغط مرتين على كل رمز.



٣ -بعد الانتهاء من الرموز يصبح البرنامج بهذا الشكل.



خصائص:

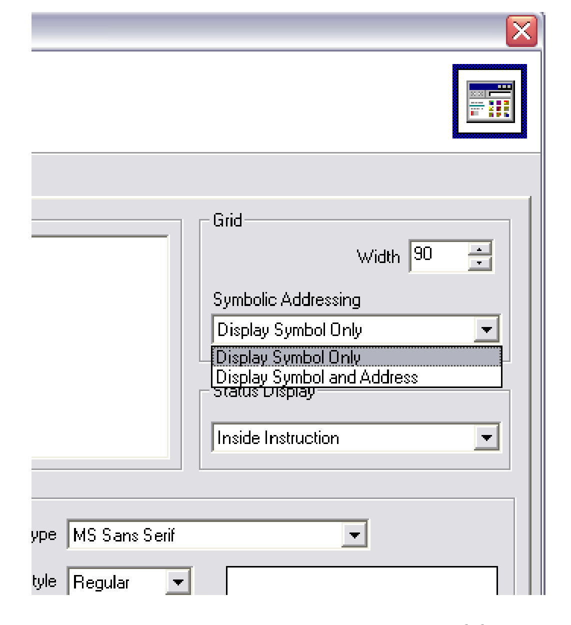
لظهور الرموز و العناوين معاً أو لظهور الرموز فقط بدون العناوين يتم الضغط على tools ثم options ثم الاختيار بين النوعين:

١ – عرض الرموز فقط

display symbol only

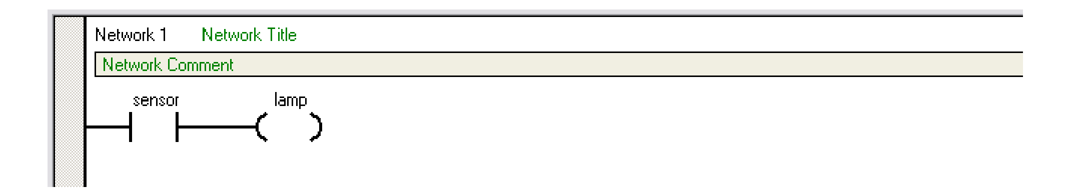
٢ – عرض الرموز والعناوين

display symbol and address



بعد الانتهاء من وضع الرموز الخاصة بكل عنوان يمكن إظهار الرموز بالطريقتين التاليتين:

- **الطريقة الأولى:** (الرموز فقط)



- **الطريقة الثانية:** (الرموز و العناوين معاً)

Network 1 Network Title
Network Comment
sensor:I0.1 lamp:Q0.0

ملاحظة:

- ويتم تطبيق الطريقة الأولى باختيار Display Symbol Only ولكن لا يفضل هذه الطريقة بسبب التخبط الذى ينتج بسببها.
- ويتم تطبيق الطريقة الثانية باختيار Display Symbol And Address وهى الطريقة الأفضل فى عرض الرموز وبالأخص فى البرامج التى تحتوى على عدد كبير من أفرع البرمجة network.
- يرجى عدم كتابة رموز تتكون من أحرف كثيرة لتجنب التخبط فى حين تطبيق الرموز على البرنامج بالفعل

ومن ضن الأعمال الهامة لصفحة "جدول الرموز" أنها تحتوى على جدول فيه مئات العناوين الخاصة بالعمليات الأكثر استخداماً بل ويوجد أيضا تعليق خاص بكل عنوان كما هى موضحاً بالصورة التالية و سوف يتم شرح بعض الأمثلة منها بالغة العربية في الفصل التاسع بالتفصيل

			Symbol	Address	Comment
1			Always_On	SM0.0	Always ON
2			First_Scan_On	SM0.1	ON for the first scan cycle only
3			Retentive_Lost	SM0.2	ON for one scan cycle, if retentive data is lost
4			RUN_Power_Up	SM0.3	ON for 1 scan cycle when RUN mode is entered from a power-up condition
5			Clock_60s	SM0.4	Clock pulse that is ON for 30 s, OFF for 30 s, for a duty cycle time of 1 min.
6			Clock_1s	SM0.5	Clock pulse that is ON for 0.5 s, OFF for 0.5 s, for a duty cycle time of 1 s.
7			Clock_Scan	SM0.6	Scan cycle clock which is ON for one cycle and OFF for the next cycle
8			Mode_Switch	SM0.7	Indicates the current position of the mode switch: 0 = TERM, 1 = RUN
9			Result_0	SM1.0	Set to 1 by the execution of certain instructions when the operation result = 0
10			Overflow_Illegal	SM1.1	Set to 1 by exec. of certain instructions on overflow or illegal numeric value.
11			Neg_Result	SM1.2	Set to 1 when a math operation produces a negative result
12			Divide_By_0	SM1.3	Set to 1 when an attempt is made to divide by zero
13			Table_Overflow	SM1.4	Set to 1 when the Add to Table instruction attempts to overfill the table
14			Table_Empty	SM1.5	Set to 1 when a LIFO or FIFO instruction attempts to read from an empty table
15			Not_BCD	SM1.6	Set to 1 when an attempt is made to convert a non-BCD value to a binary value
16			Not_Hex	SM1.7	Set to 1 when an ASCII value cannot be converted to a valid hexadecimal value
17			Parity_Err	SM3.0	Set to 1 if a parity error is detected in a char received by Port 0 or Port 1
18			Comm_Int_Ovr	SM4.0	Set to 1 if the communication interrupt queue overflows (interrupt routine only)
19			Input_Int_Ovr	SM4.1	Set to 1 if the input interrupt queue overflows (interrupt routine only)
20			Timed_Int_Ovr	SM4.2	Set to 1 if the timed interrupt queue overflows (interrupt routine only)

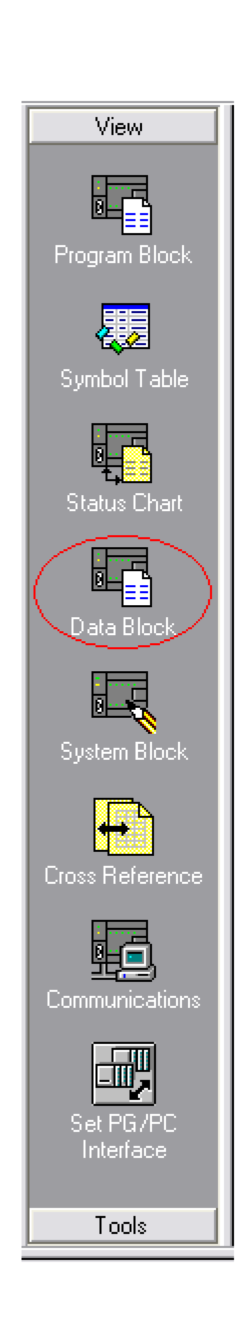
S7-200 Symbols | USER2 | POU Symbols

الباب الرابع

صفحة البيانات

- شرح صفحة البيـــــــــــــــــــــــــــــانات.
- محتويات صفحة البيـــــــــــــــــــــــــانات.
- المفاتيح المستخدمة فى صفخة البيـــــــــــــــانات.
- الأخـــــــــــــــــــــــــطاء الممكن التعرض لها.
- التعليقات الخاصة بصفحة البيـــــــــــــــــانات.
- كيفية تخزين قيم مسبقة على متغيـــــرات بحجم Bit.
- كيفية تخزين قيم مسبقة على متغيـــرات بحجم Byte.
- كيفية تخزين قيم مسبقة على متغيـــرات بحجم Word.
- كيفية تخزين قيم مسبقة على متغيرات بحجم Dword.
- تمـــــــــــــــــــــــــــــــارين عملية للتوضيح.

صفحة البيانات Data Block :



تستخدم صفحة " البيانات" الـ Data Block كصفحة للتعليقات أو كصفحة لتحديد قيم مسبقة خاصة بالمتغيرات بجميع أنواعها سواء المستخدمة فى البرنامج أو غير المستخدمة فى البرنامج وهذا لأن فى جميع الحالات تكون القيمة الأولية للمتغيرات هى صفر وهذا قد يسبب بعض المشاكل وهذا يحدث أذا تم استخدام المتغيرات بالتحديد مع المؤقتات الزمنية, مع العدادات أو حتى مع مفاتيح المقارنة كما سوف يتم التوضيح فى التمارين العملية فى نفس الفصل.

استخدامات صفحة " البيانات":

– استخدام صفحة البيانات للتعليقات:

يمكن كتابة التعليقات قبل أو بعد تحميل البرنامج و هذا لأن التعليقات لا تعنى أى شيء بالنسبة لجهاز الـ PLC, فمثلاً لن يؤثر على البرنامج أذا كتبت كتعليق أن ثمن المفتاح هو عشرة جنيهات أو أن لون الموتور أزرق.

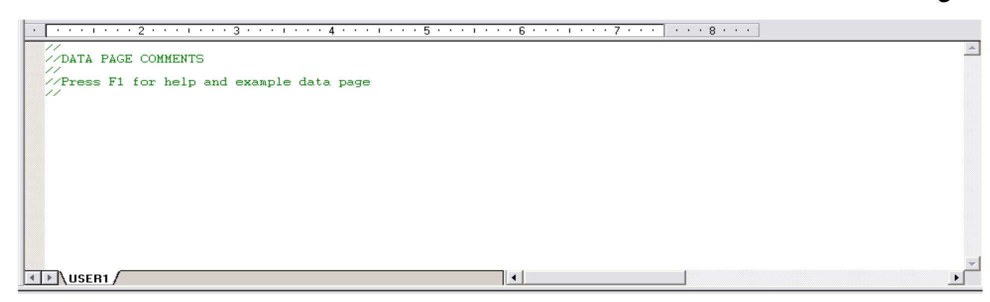
- استخدام صفحة البيانات للمتغيرات:

يجب كتابة القيم الخاصة بالمتغيرات قبل تحميل البرنامج و ليس بعد و هذا لأن تلك القيم تعنى الكثير لجهاز ال PLC لأنها ستؤثر على البرنامج, فمثلاً أذا كتبت أن قيمة المتغير VW0 هى ٤٠٠٠ فمثلاً أذا كان هذا المتغير مستخدم مع مؤقت زمنى T32 فهذا يعنى أن المؤقت الزمنى سوف يغير النقط الخاصة به بعد أربع ثوانى من التشغيل.

ملاحظة:

- فى حالة عدم كتابة أى قيمة مسبقة للمتغيرات فهذا يعنى أن القيمة الحالية للمتغيرات ستكون صفر و قد يتسبب هذا فى بعض المشاكل.
- يمكن تغير قيمة أى متغير بواسطة صفحة "جدول الحالات" حتى ولو كان لهذا المتغير قيمة مسبقة, ويتم هذا بواسطة أمر write all أنظر صفحة ٤٢.
- لكتابة بعض التعليقات يجب إن تكتب فى البداية "//" ثم يكتب التعليق المراد كتابته.
- لكتابة بعض المتغيرات يجب إن لا تكتب فى البداية "//" بل يكتب المتغير المراد كتابته دون أى مقدمات.
- فى حالة كتابة أى متغيرات بطريقة خطاء أو كتابة أى تعليق دون "//" تظهر تلقائيا علامة "X" لكى تشير إلى أنه قد تم كتابة شىء غير صحيح.
- الطريقة الأفضل لتغيير محتويات المتغيرات بواسطة البرنامج تتم عن طريق عمليات النقل, أنظر صفحة ٢٠٩ فى الجزء الأول من الكتاب.

شكل صفحة البيانات:



المفاتيح المستخدمة فى صفحة البيانات.



- Compile all: بالضغط على هذا المفتاح يقوم البرنامج بمراجعة صفحة البيانات لإظهار الأخطاء أذا وجدت.
- Download: بالضغط على هذا المفتاح يقوم بتحميل البرنامج كما سبق و تحميل صفحة البيانات أيضاً.

مثال عملى باستخدام صفحة "جدول المرجع":

محرك يعمل يميناً لوقت ثم يقف لوقت ثم يعمل تلقائياً لليسار لوقت أخر ثم يقف لوقت و هكذا بشرط أن يكون هناك أمكانية لتغير الزمن.

عدد الدخل	نوع الدخل	أسم الدخل
١	n.c.	I0.0/S1
٢	n.o.	I0.1/S2
عدد المتغيرات	**نوع المتغيرات**	**أسم المتغيرات**
١	<=I	T200/VW0

٢	>=I	T200/ VW2
٣	<=I	T200/VW4
٤	<=I	T200/ VW6
عدد المتغيرات	**نوع المتغيرات**	**أسم المتغيرات**
١	Word	VW0
٢	Word	VW2
٣	Word	VW4
٤	Word	VW6
عدد الريليهات	**نوع الريليهات**	**أسم الريليهات**
١	Bit	M0.0
عدد المؤقتات الزمنية	**نوع المؤقتات الزمنية**	**أسم المؤقتات الزمنية**
١	TON	T200
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتكتور	Q0.1/K1M
٢	كونتكتور	Q0.2/K2M

الشرح:

Network1: بالضغط على I0.1 يعمل الريليه M0.0 بشرط أن يكون I0.0 مغلق.

Network2: يعمل المؤقت الزمنى T96 لزمن متغير بقيمة مسبقة ثانيتان ثم يقف ليبدأ من جديد.

Network3: يعمل المحرك يميناً لزمن متغير بقيمة مسبقة ثانيتان ثم يقف لزمن متغير بقيمة مسبقة ثانيتان أيضاً.

Network4: يعمل المحرك يساراً لزمن متغير بقيمة مسبقة ثانيتان ثم يقف لزمن متغير بقيمة مسبقة ثانيتان أيضاً.

البرنامج:

Network 1 Network Title

Network Comment

I0.1 I0.0 M0.0

M0.0

Network 2

M0.0 T200 T200

<=I IN TON

VW6

12 PT 100 ms

Network 3

M0.0 T200 Q0.1

<=I

VW0

Network 4

M0.0 T200 T200 Q0.2

>=I <=I

VW2 VW4

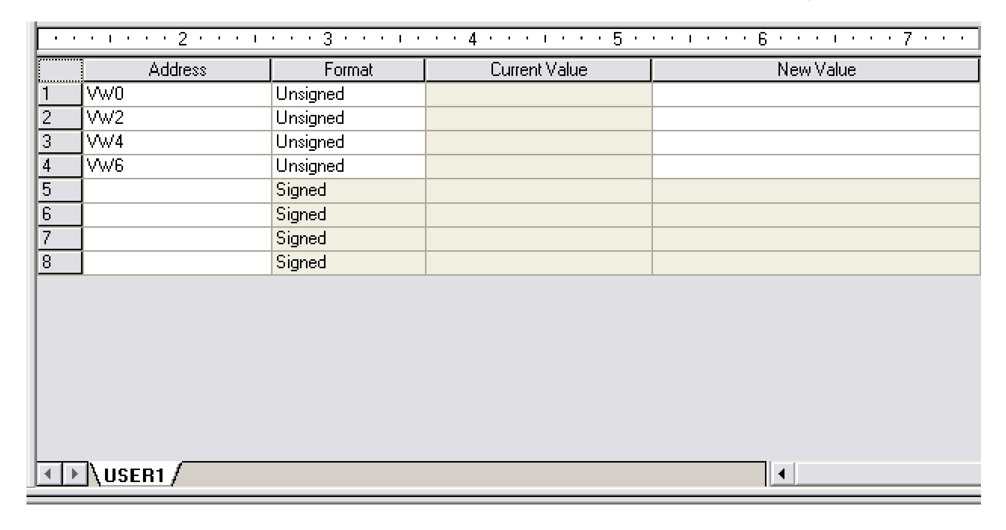
تم تحديد قيم المتغيرات بواسطة صفحة البيانات كما سبق وشرحنا.

```
//
//DATA PAGE COMMENTS
//
//Press F1 for help and example data page
//


VW0     20                            // VARIABLE MEMORY VW0(SIZE WORD).
VW2     40                            // VARIABLE MEMORY VW2(SIZE WORD).
VW4     60                            // VARIABLE MEMORY VW4(SIZE WORD).
VW6     80                            // VARIABLE MEMORY VW6(SIZE WORD).
```

USER1

بل يمكن أيضاً تغير قيم المتغيرات مرة أخرى بواسطة صفحة "جدول الحالات" بواسطة أمر write all.



	Address	Format	Current Value	New Value
1	VW0	Unsigned		
2	VW2	Unsigned		
3	VW4	Unsigned		
4	VW6	Unsigned		
5		Signed		
6		Signed		
7		Signed		
8		Signed		

USER1

بالضغط على مفتاح Compile all يقوم البرنامج بمراجعة صفحة البيانات لإظهار الأخطاء أذا وجدت.

```
Compiling Program Block...
MAIN (OB1)
SBR_0 (SBR0)
INT_0 (INT0)
Block Size = 64 (bytes), 0 errors

Compiling Data Block...
Block Size = 8 (bytes), 0 errors

Compiling System Block...
Compiled Block with 0 errors, 0 warnings

Total Errors: 0
```

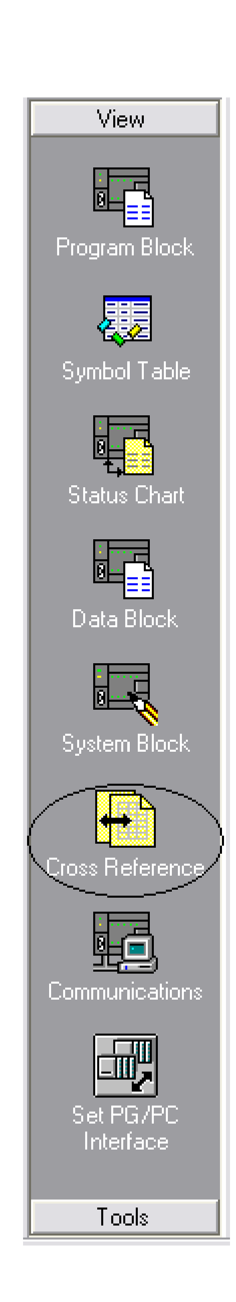
Ready

الباب الخامس

جدول المرجع

- شرح جدول المـــــــــــــــــرجع.
- محتويات جدول المـــــــــــــــرجع.
- صـــــــــــفحة Cross Reference.
- صـــــــــــــــفحة Byte Usage.
- صـــــــــــــــفحة Bit Usage.
- المفاتيح المستخدمة فى جدول المرجع.
- الأخـــــــــــطاء الممكن التعرض لها.
- تمـــــــــــــــارين عملية للتوضيح.

جدول المرجع Cross Reference :



تستخدم صفحة "جدول المرجع" الـ Cross Reference لكى تستخدم كمرجع لكل العناوين المستخدمة فى البرنامج بجميع أنوعها. حيث تنقسم هذه الصفحة إلى ثلاث صفحات فرعية:

الصفحة الفرعية الأولى تستخدم لعرض جميع العناوين مثل: المداخل, المخارج, الريليهات, المتغيرات, العدادات, المؤقتات, الخ.....

الصفحة الفرعية الثانية تستخدم لعرض بعض العناوين مثل: المتغيرات, العدادات و المؤقتات.

الصفحة الفرعية الثالثة تستخدم لعرض عناوين بحجم bit مثل: المداخل, المخارج و الريليهات.

طرق استخدام صفحة "جدول المرجع":

– يتم عمل البرنامج أولاً ثم تحميل البرنامج ثم فتح صفحة "جدول المرجع".

– يتم عمل البرنامج أولاً ثم الضغط على compile ثم فتح صفحة "جدول المرجع".

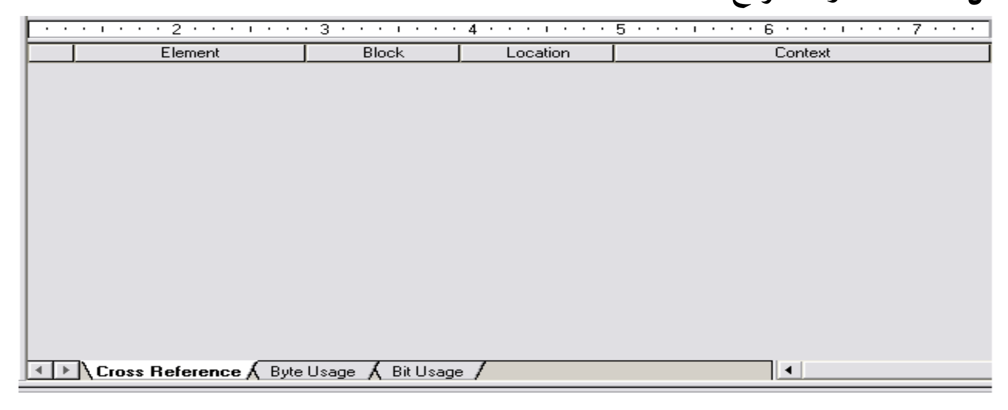
ملاحظة:

فى حاله رسم البرنامج ثم فتح صفحة "جدول المرجع" مباشر دون تحميل البرنامج أو الضغط على compile تظهر هذه الرسالة:

A compile must be performed to display cross reference.

أى أنه يجب الضغط على compile لإظهار صفحة "جدول المرجع".

شكل صفحة "جدول المرجع":



الصفحة الفرعية الأولى Cross Reference تحتوى على:

Element – Block – Location – Contexet

١–Element: حيث تظهر كل العناوين التى استخدمت فى البرنامج و أذا قد استخدم العنوان أكثر من مرة فأنه يظهر بنفس عدد المرات التى تكرر بها فى البرنامج.

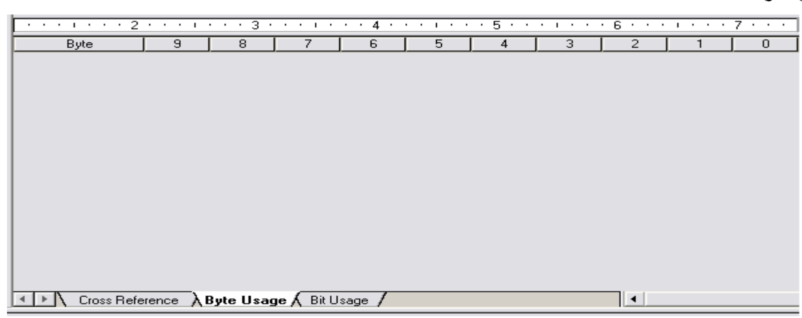
٢–Block: حيث يظهر لك فى أى صفحة برمجة (MAIN أو SBR أو INT) يوجد هذا العنوان.

۳–Location: حيث يشير فى أى فرع من فروع (network 1 أو network 2 أو .....) البرنامج قد استخدم هذا العنوان.

٤-Context: حيث يضيف بعض التوضيحات الخاصة بالعنوان, فمثلاً أذا كان العنوان مفتاح فأنه يوضح أذا كان مفتوح أو مغلق أو مفتاح مقارنة, أو أذا كان العنوان هو لمؤقت زمنى فأنه يوضح أذا كان نوعه TON أو TOF أو TONR, أو أذا كان العنوان هو لعداد فأنه يوضح أذا كان نوعه CTU أو CTD أو CTUD وهكذا....

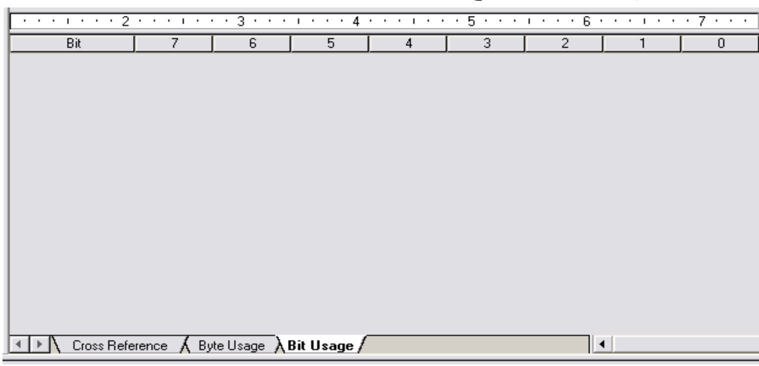
الصفحة الفرعية الثانية Byte Usage:

تستخدم لإظهار أسم العدادات و المؤقتات الزمنية و يشير أيضاً إلى مجموعة الـ byte المستخدمة سواء للمتغيرات أو للريليهات



الصفحة الفرعية الثالثة Bit Usage:

حيث يستخدم لإظهار أسم المداخل, المخارج أو الريليهات ولكن كل عنوان على حدا



حيث يستخدم لإظهار أسم العناوين التى هي مثل: الدخل, الخرج, الريليه أو المتغير. أى أنه يشير إلى عناوين بحجم bit.

المفاتيح المستخدمة فى صفحة الرموز.



Compile: بالضغط على هذا المفتاح تصبح صفحة "جدول المرجع" متاحة.

Compile all بالضغط على هذا المفتاح أيضاً تصبح صفحة "جدول المرجع" متاحة.

Download: بالضغط على هذا المفتاح يقوم بتحميل البرنامج كما سبق و شرحنا و تصبح صفحة "جدول المرجع" متاحة.

مثال عملى باستخدام صفحة "جدول المرجع":

محرك يعمل يميناً لوقت ثم يقف لوقت ثم يعمل تلقائياً لليسار لوقت أخر ثم يقف لوقت و هكذا.

عدد الدخل	نوع الدخل	أسم الدخل
١	n.c.	I0.0/S1
٢	n.o.	I0.1/S2
عدد المتغيرات	**نوع المتغيرات**	**أسم المتغيرات**
١	>=I	T96
٢	<=I	T96
٣	<=I	T96

٤	<>I	T96
عدد الريلهات	**نوع الريلهات**	**أسم الريلهات**
١	Bit	M0.0
عدد المؤقتات	**نوع المؤقتات**	**أسم المؤقتات**
١	TON	T96
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتكتور	Q0.1/K1M
٢	كونتكتور	Q0.2/K2M

البرنامج:

Network 1 Network Title
Network Comment
I0.1 I0.0 M0.0
M0.0

Network 2
M0.0 T96 <>I 8000 T96 IN TON
2 PT 1 ms

Network 3
M0.0 T96 <=I 2000 Q0.1

Network 4
M0.0 T96 >=I 4000 T96 <=I 6000 Q0.2

الشرح:

Network1:

بالضغط على I0.1 يعمل الريليه M0.0 بشرط أن يكون I0.0 مغلق.

Network2:

يعمل المؤقت الزمني T96 لمدة ثمان ثواني ثم يقف ليبدأ من جديد.

Network3:

يعمل المحرك يميناً لمدة ثانيتان (من صفر إلى أثنين) ثم يقف لمدة ثانيتان أخريتان (من أثنين إلى أربعة).

Network4:

يعمل المحرك يساراً لمدة ثانيتان (من أربعة إلى ستة) ثم يقف لمدة ثانيتان أخريتان (من ستة إلى ثمانية).

صفحة Cross Reference الخاصة بهذا البرنامج:

	Element	Block	Location	Context
1	I0.0	MAIN (OB1)	Network 1	-| |-
2	I0.1	MAIN (OB1)	Network 1	-| |-
3	Q0.1	MAIN (OB1)	Network 3	-( )
4	Q0.2	MAIN (OB1)	Network 4	-( )
5	M0.0	MAIN (OB1)	Network 1	-( )
6	M0.0	MAIN (OB1)	Network 1	-| |-
7	M0.0	MAIN (OB1)	Network 2	-| |-
8	M0.0	MAIN (OB1)	Network 3	-| |-
9	M0.0	MAIN (OB1)	Network 4	-| |-
10	T96	MAIN (OB1)	Network 2	-|<>I|-
11	T96	MAIN (OB1)	Network 2	TON
12	T96	MAIN (OB1)	Network 3	-|<=I|-
13	T96	MAIN (OB1)	Network 4	-|>=I|-
14	T96	MAIN (OB1)	Network 4	-|<=I|-

Cross Reference / Byte Usage / Bit Usage

صفحة Byte Usage الخاصة بهذا البرنامج:

Byte	9	8	7	6	5	4	3	2	1	0
MB0										b
T0										
T10										
T20										
T30										
T40										
T50										
T60										
T70										
T80										
T90				X						

Cross Reference | **Byte Usage** | Bit Usage

ملاحظة:

⇐ تحتوى هذه الصفحة على مسميات المؤقتات الزمنية المستخدمة فى البرنامج, بحيث يكتب الاسم بطريقة خاصة فتكتب مسميات المؤقتات بداياً من صفر إلى أن يصل إلى أسم المؤقت المستخدم فعلاً و تشير أيضاً إلى أنه أستخدم الريليه الأول.

⇐ فى حالة استخدام ذاكرة "متغيرات" بحجم word مثلاً فأنه يتم الإشارة إلى الـ bytes المكونة لهذا الـ word فمثلاً أذا كان المتغير المستخدم فى البرنامج هو VW4 فأنة يشير إلى هذه الذاكرة داخل صفحة "جدول المرجع" بواسطة كتابة حرف الـ W أمام الـ byte4 والـ byte5 ليشير إلى أنة تم استخدام الـ byte4 والـ byte5 ولكن عن طريق الـ word4.

⇐ فى حالة استخدام ذاكرة "متغيرات" بحجم Dword مثلاً فأنه يتم الإشارة إلى ال bytes المكونة لهذا ال Dword فمثلاً أذا كان المتغير المستخدم فى البرنامج هو VD0 فأنة يشير إلى هذه الذاكرة داخل صفحة "جدول المرجع" بواسطة كتابة حرف ال D أمام ال byte0 وال byte1 وال byte2 وال byte3 ليشير إلى أنة تم استخدام كل هذه ال bytes ولكن عن طريق ال Dword0.

صفحة Bit Usage **الخاصة بهذا البرنامج:**

Bit	7	6	5	4	3	2	1	0
I0.0							b	b
Q0.0						b	b	
M0.0								b

Cross Reference / Byte Usage / **Bit Usage**

هذه الصفحة تشير إلى كل العناوين التى استخدمت فى البرنامج بشرط أن تكون بحجم bit, مثل:

I0.0, I0.1, Q0.1, Q0.2 و M0.0

الباب السادس

البرامج الفرعية

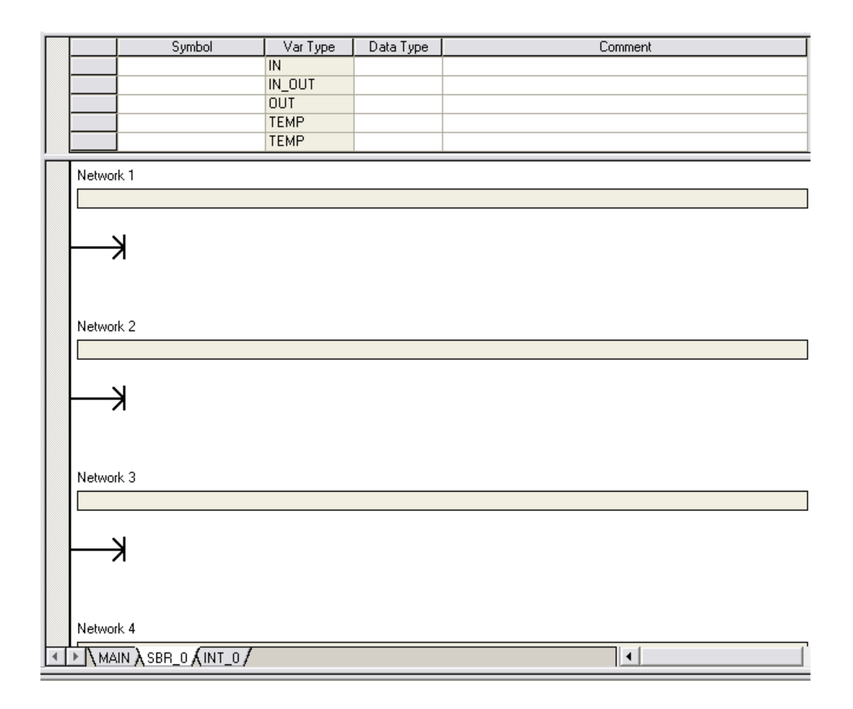
- شرح البرامـــــــــــــــــــــج الفرعية.
- محتويات البرامـــــــــــــــــــج الفرعية.
- الجدول المستخدم فى البرامـــــــــــج الفرعية.
- الأخطاء الممكن حدوثها فى جدول البرامج الفرعية.
- التعليقات الخاصة بصفحة البرامـــــــج الفرعية.
- جدول الرموز الخاص بصفحات البرمجـــــــــــة.

صفحة البرامج الفرعية Subroutine :

تستخدم طريقة "البرامج الفرعية" الـ Subroutine فى:

١- البرامج الكبيرة التى تحتوى على أجزاء متكررة.

٢- لتقسيم البرنامج الكبير إلى أجزاء صغيرة لسهولة حل الأعطال.



الحالة الأولى:

- فى حاله وجود نظام الإنذارات الخاص بكل مكينة فأن الإنذارات تعمل بنفس الطريقة حتى مع اختلاف الماكينات .
- فى حالة وجود لمبات أشارة تضىء مع كل محرك فأن لمبات تعمل بنفس الطريقة حتى مع اختلاف المحركات.

- فى حاله وجود فلاشر يضاء حسب شروط خاصة فأن الفلاشر يعمل بنفس الطريقة حتى مع اختلاف الشروط .

ملاحظة:

أى أنة مثلاً بدلاً من تكرار أى جزء عشرة مرات فأنة يرسم مرة واحدة فقط ولكن يتم أعطاء هذا الجزء عشرة عناوين مختلفة كما لو كان مرسوم عشرة مرات فعلاً.

الحالة الثانية:

- فى حاله وجود برنامج لمجموعة ماكينات مختلفة فأنة يفضل استخدام صفحات مختلفة لكل مكينة حيث أنه بتقسيم البرنامج يسهل على المبرمج التوصل لسبب الأعطال بسهولة وفى وقت قصير.

طريقة استخدام صفحة "البرامج الفرعية":

- **أولاً** يتم رسم الجزء الذى كان من المفروض أن يتكرر داخل صفحة " البرامج الفرعية " ولكن مرة واحدة فقط ودون أعطاء أى عناوين محددة.

	Symbol	Var Type	Data Type	Comment
		IN		
		IN_OUT		
		OUT		
		TEMP		
		TEMP		

Network 1 Network Title

Network Comment

??.? ??.? ??.?

- **ثانياً** يتم تحديد كل المسميات التى سوف تستخدم فى البرنامج الفرعى داخل الجدول الخاص بالبرامج الفرعية مع تحديد أذا كان هذا الاسم هو لدخل أو لخرج أو ....

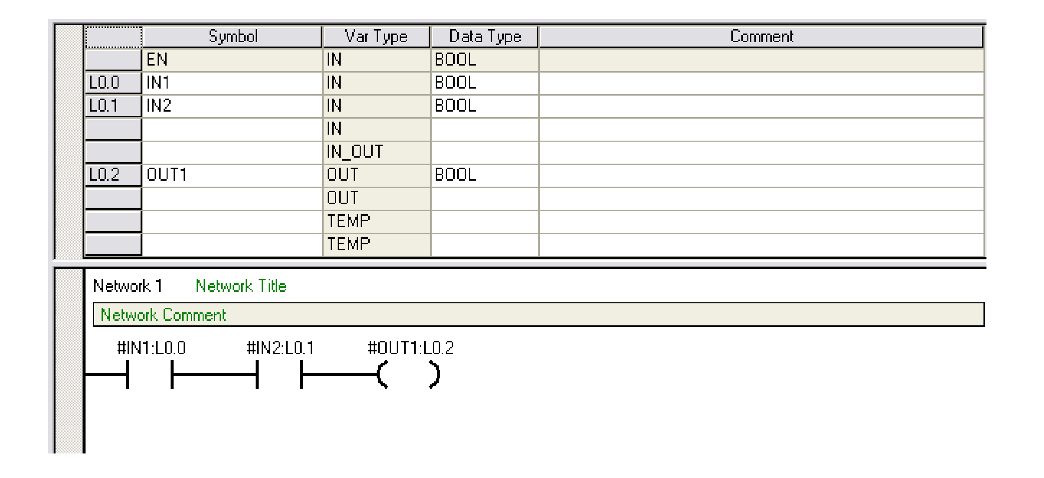
	Symbol	Var Type	Data Type	Comment
	EN	IN	BOOL	
L0.0	IN1	IN	BOOL	
L0.1	IN2	IN	BOOL	
		IN		
		IN_OUT		
L0.2	OUT1	OUT	BOOL	
		OUT		
		TEMP		
		TEMP		

الجدول الخاص بالبرامج الفرعية.

جدول الرموز Var Type الخاص بصفحة البرامج الفرعية يحتوى على:

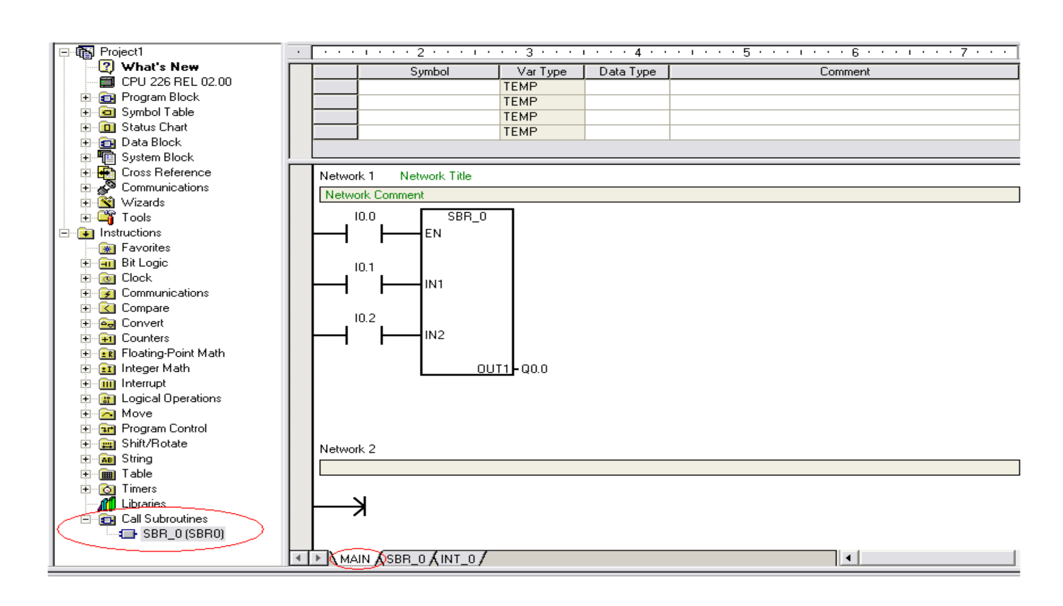
- IN: أى أنه يستخدم فى البرامج الفرعية كدخل.
- OUT: أى أنه يستخدم فى البرامج الفرعية كخرج.
- IN_OUT: أى أنه يستخدم فى البرامج الفرعية كدخل و خرج مثل الريله.
- TEMP: أى أنه يستخدم فى البرامج الفرعية مثل المتغيرات
- BOOL: تستخدم مع عناوين بحجم bit.
- BYTE: تستخدم مع عناوين بحجم byte.
- WORD: تستخدم مع عناوين بحجم word(unsigned).
- INT: : تستخدم مع عناوين بحجم word(signed).
- DWORD: تستخدم مع عناوين بحجم Dword(unsigned).
- DINT: تستخدم مع عناوين بحجم Dword(signed).
- REAL: تستخدم مع عناوين بحجم Dword(real).
- STRING: تستخدم مع عناوين بحجم byte.

- **ثالثآ** يتم كتابة الأسم الذى تم تحديدة فى البرنامج بدل العنوان و الهدف من هذا أنة أذا كتبنا عنوان فهذا يعنى أنة تم اختيار العنوان إلى الأبد ولكن فى حاله كتابة أى رمز فأنة يمكن بكل سهوله فى ما بعد أعطاء عناوين مختلفة لنفس الرمز.



	Symbol	Var Type	Data Type	Comment
	EN	IN	BOOL	
L0.0	IN1	IN	BOOL	
L0.1	IN2	IN	BOOL	
		IN		
		IN_OUT		
L0.2	OUT1	OUT	BOOL	
		OUT		
		TEMP		
		TEMP		

- **رابعاً** يتم الاتصال بصفحة "البرامج الفرعية" عن طريق أمر CALL_SUB الذى يوجد فى صفحة "البرنامج الرئيسية" مع تحديد العنوان الذى سيحل محل الرمز.



الأخطاء الممكن حدوثها فى جدول البرامج الفرعية:

١– ممنوع كتابة أى رمز يكون بالصدفة أسم من أسماء العماليات المستخدمة.

- أى أنه لا يمكن كتابة كلمة ("stop", "S", "R", "JMP", "LBL", "END", "ENI", "TON", "TOF", "TONR", "CTU", "CTD", "CTUD") حيث أن كل هذه الكلمات السابقة تمثل عمليات تستخدم فى جهاز ال PLC.

٢– ممنوع كتابة أى رمز مكون من كلمتين و يكون بينهم مسافة.

- أى أنه لا يمكن كتابة كلمة ("first button", "second button ", "motor right", "motor left") حيث أن المسميات المكتوبة هى مكونة من كلمتين ولكن لحل هذه المشكلة يتم وضع أى رمز أو علامة بين الكلمة الأولى والكلمة الأخرى.

٣– ممنوع تكرار نفس الرمز مع عناوين مختلفة.

- أى أنه لا يمكن كتابة أى رمز أكثر من مرة حتى ولو كان مع نفس العنوان حيث أن فى هذه الحالة يصعب على ال PLC وضع رمزين مع نفس العنوان فيترك العنوان بدون رمز .

٤– اختلاف بين الرموز المحددة فى الجدول و المستخدمة فى البرنامج الفرعى.

- أى أنه لا يمكن كتابة أى أسم فى الجدول بحروف كبيرة بينما يكتب فى البرنامج الفرعى بحروف صغيرة أو حتى أن يعرف العنوان فى الجدول بأسم IN1 بينما يستخدم فى البرنامج الفرعى بأسم in1 لأن بالنسبة لوحدة البرمجة ال PLC يوجد فرق كبير بين الأسمين.

المفاتيح المستخدمة فى صفحة البرامج الفرعية



١- Program Status:

بالضغط على هذا المفتاح تظهر الحالة الخاصة بكل عنوان.

٢- Compile all:

بالضغط على هذا المفتاح تظهر عدد الأعطال الخاصة بالبرنامج.

مثال عملى:

محركين يعمل كل محرك فى اتجاهين بحيث أن مع كل أتجاة تعمل لمبتان كفلاشر بشرط أن اللمبات الخاصة بالأتجاة اليمين مختلفة عن اللمبات الخاصة بالاتجاه اليسار و اللمبات الخاصة بالمحرك الأول مختلفة عن المحرك الثانى.

المداخل

عدد الدخل	نوع الدخل	أسم الدخل
١	n.c.	I0.0/S1
٢	n.o.	I0.1/S2
٣	n.o.	I0.2/S3
٤	n.o.	I0.3/S4
٥	n.o.	I0.4/S5
٦	n.o.	I1.1/S6
٧	n.o.	I1.2/S7
٨	n.o.	I1.3/S8
٩	n.o.	I1.4/S9

المؤقتات الزمنية

عدد المؤقتات الزمنية	نوع المؤقتات الزمنية	أسم المؤقتات الزمنية
١	TON	T32
٢	TON	T96

المخارج

عدد الخرج	نوع الخرج	أسم الخرج
١	كونتكتور	Q0.1/K1M
٢	كونتكتور	Q0.2/K2M
٣	كونتكتور	Q0.3/K3M
٤	كونتكتور	Q0.4/K4M
٥	لمبة	Q1.1/K5M
٦	لمبة	Q2.1/K6M
٧	لمبة	Q1.2/K7M
٨	لمبة	Q2.2/K8M
٩	لمبة	Q1.3/K9M
١٠	لمبة	Q2.3/K10M
١١	لمبة	Q1.4/K11M
١٢	لمبة	Q2.4/K12M

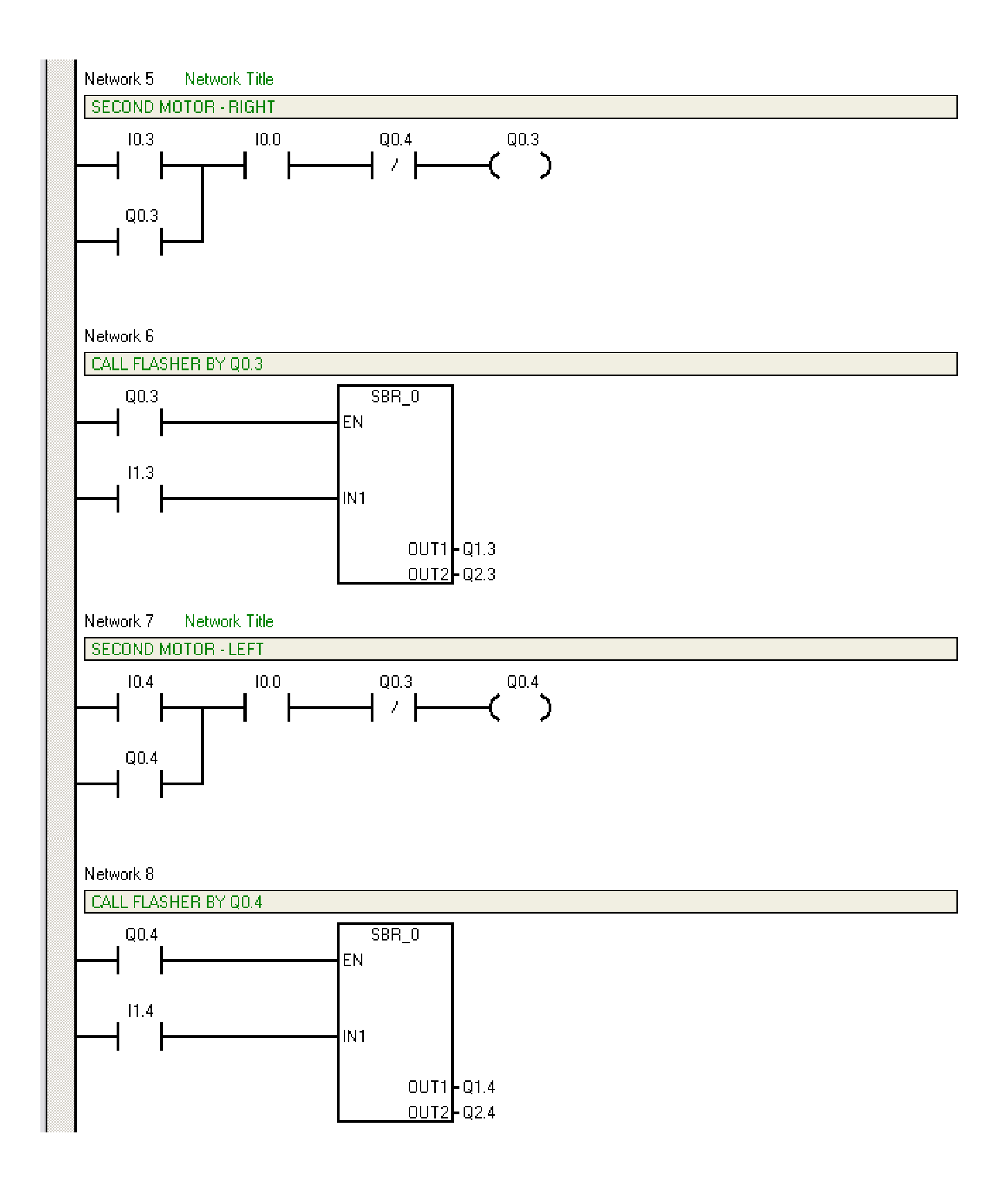
البرنامج الرئيسى:

Network 1 Network Title
FIRIST MOTOR - RIGHT
I0.1 I0.0 Q0.2 / Q0.1 ( )
Q0.1

Network 2
CALL FLASHER BY Q0.1
Q0.1 SBR_0 EN
I1.1 IN1
OUT1 Q1.1
OUT2 Q2.1

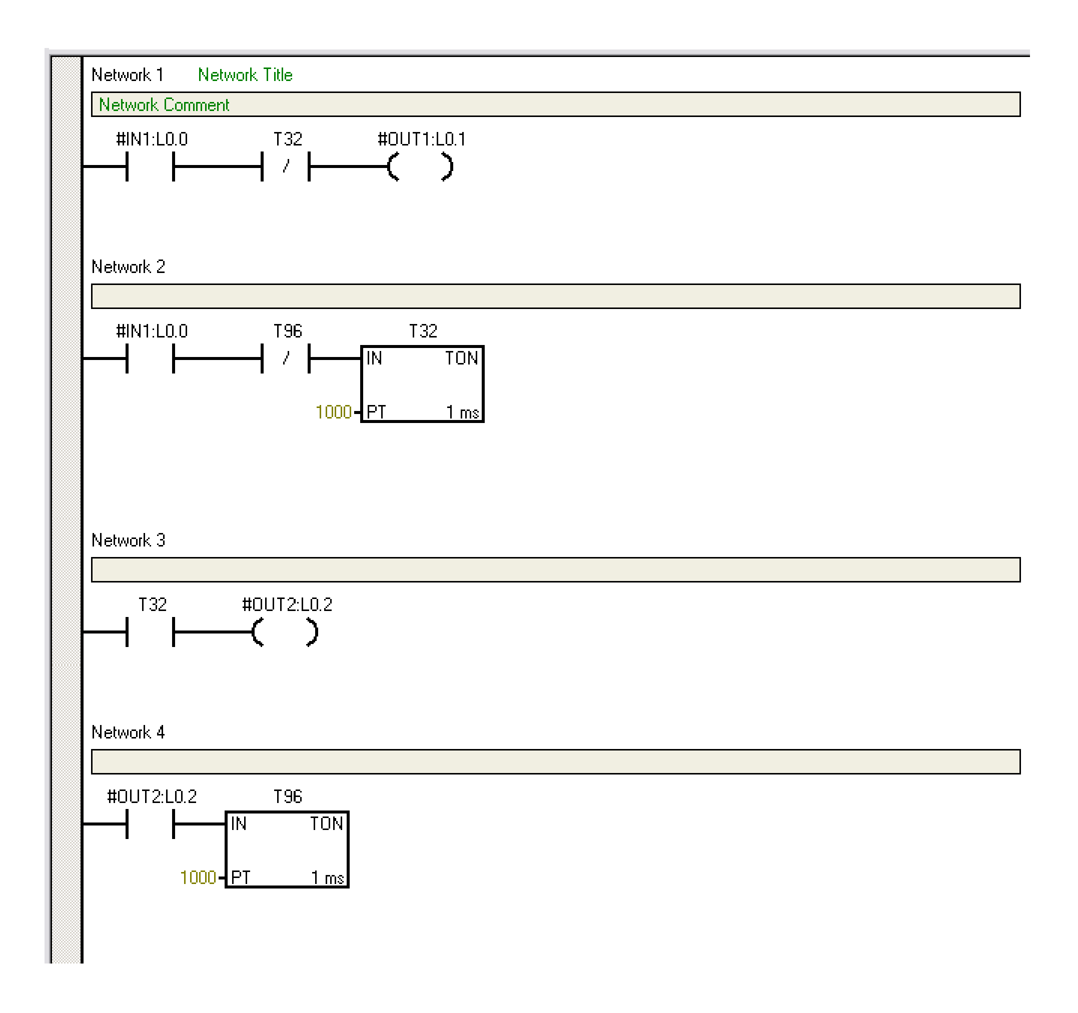
Network 3 Network Title
FIRIST MOTOR - LEFT
I0.2 I0.0 Q0.1 / Q0.2 ( )
Q0.2

Network 4
CALL FLASHER BY Q0.2
Q0.2 SBR_0 EN
I1.2 IN1
OUT1 Q1.2
OUT2 Q2.2



Network 5
Network Title
SECOND MOTOR - RIGHT
I0.3
I0.0
Q0.4
Q0.3
Q0.3
Network 6
CALL FLASHER BY Q0.3
Q0.3
SBR_0
EN
I1.3
IN1
OUT1
Q1.3
OUT2
Q2.3
Network 7
Network Title
SECOND MOTOR - LEFT
I0.4
I0.0
Q0.3
Q0.4
Q0.4
Network 8
CALL FLASHER BY Q0.4
Q0.4
SBR_0
EN
I1.4
IN1
OUT1
Q1.4
OUT2
Q2.4

البرنامج الفرعى:



- **شرح البرنامج الرئيسى:**

Network1: الجزء الخاص بحركة المحرك الأول لليمين

Network2: الجزء الخاص بالإتصال بالفلاشر الموجود فى البرنامج الفرعى الخاص بحركة المحرك الأول لليمين.

Network3: الجزء الخاص بحركة المحرك الأول لليسار

Network4: الجزء الخاص بالإتصال بالفلاشر الموجود فى البرنامج الفرعى الخاص بحركة المحرك الأول لليسار.

Network5: الجزء الخاص بحركة المحرك الثانى لليمين

Network6: الجزء الخاص بالإتصال بالفلاشر الموجود فى البرنامج الفرعى الخاص بحركة المحرك الثانى لليمين.

Network7: الجزء الخاص بحركة المحرك الثانى لليسار

Network8: الجزء الخاص بالإتصال بالفلاشر الموجود فى البرنامج الفرعى الخاص بحركة المحرك الثانى لليسار.

- **شرح البرنامج الفرعى:**

Network1: الجزء الخاص باللمبة الأولى.

Network2: الجزء الخاص بالمؤقت الزمنى T32.

Network3: الجزء الخاص باللمبة الثانية.

Network4: الجزء الخاص بالمؤقت الزمنى T96.

ملاحظة.

باستخدام الـ subroutine أصبح عدد الفروع ١٢بدلاً من ٢٠

الباب السابع

البوابات

- شرح البوابـــــــــــــــــات.
- بوابـــــــــــــــــات "و".
- بوابـــــــــــــــــات "أو".
- بوابـــــــــــــــــات "نفى".
- أحجام البوابـــــــــــــــــات.
- المفاتيح المستخدمة مع البوابات.
- الأخـــــطاء الممكن التعرض لها.
- توضيحات بالرســـــــــــــــم.
- تمـــــــــــــــــارين عملية.

البوابات المنطقية:

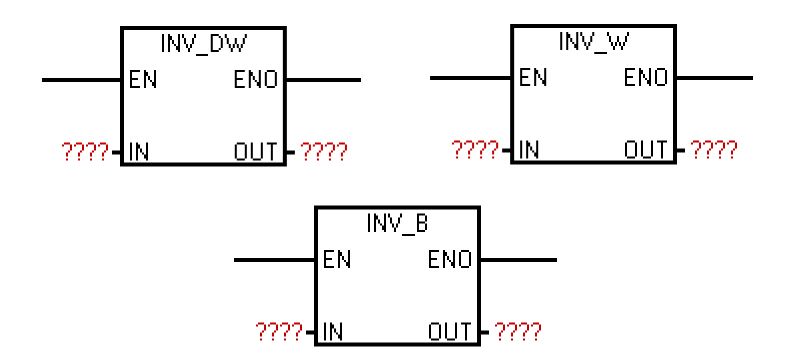
تستخدم البوابات فى البرامج العادية أو حتى التى تحتوى على عمليات حسابية.

أنواع البوابات:

,INV_B ,INV_W ,INV_DW ,WAND_B ,WAND_W ,WAND_DW ,WOR_B ,WOR_W ,WOR_DW ,WXOR_B ,WXOR_W ,WXOR_DW.

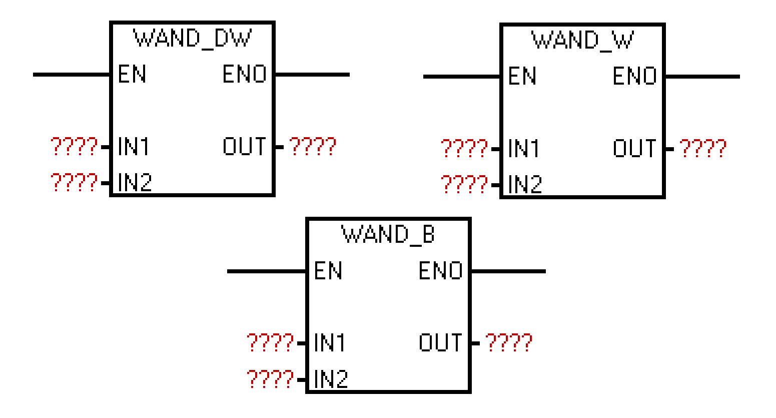
بوابات النفى "INVERT":

تقوم بتحويل الصفر إلى واحد و الواحد إلى صفر و تتواجد هذه العملية بحجم Byte, Word و Dword.



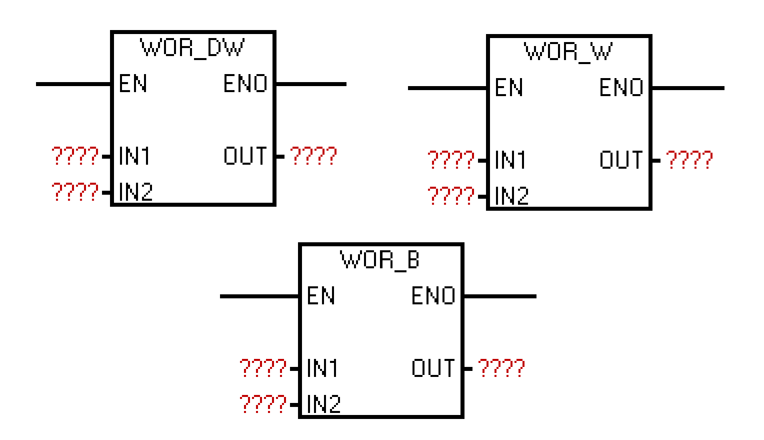
بوابات "AND":

تقوم بتطبيق البوابة المنطقية "و" بين مجموعتين و تتواجد هذه العملية بحجم Byte, Word و Dword.



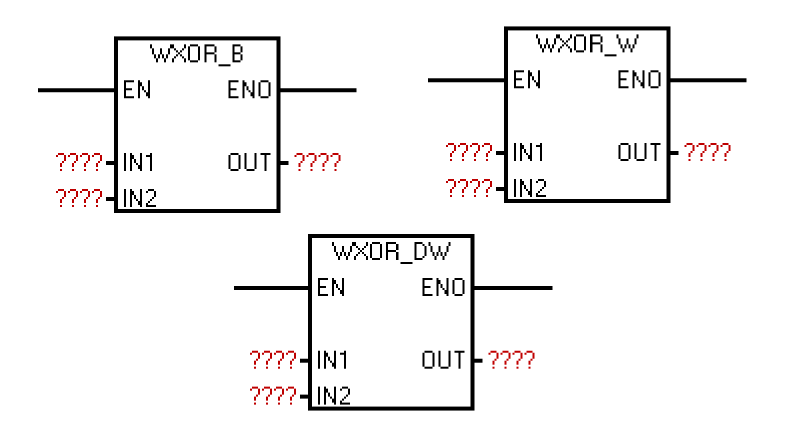
بوابات "OR":

تقوم بتطبيق البوابة المنطقية "أو" بين مجموعتين و تتواجد هذه العملية بحجم Byte, Word و Dword



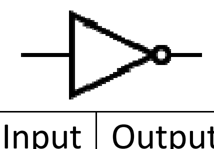
بوابات "XOR":

تقوم بتطبيق البوابة المنطقية "XOR" بين مجموعتين و تتواجد هذه العملية بحجم Byte, Word و Dword.



شرح البوابات:

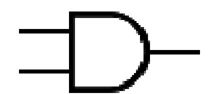
بوابات INV:



Input	Output
A	INV(A)
1	0
0	1

البوابة "نفى" تقوم بتحويل الواحد إلى صفر و الصفر إلى واحد.

بوابات AND:



Input		Output
A	B	A x B
0	0	0
0	1	0
1	0	0
1	1	1

البوابة "و": تصبح القيمة واحد عندما تتحقق كل القيم.

بوابات OR:



Input		Output
A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	1

البوابة "أو": تصبح القيمة واحد عندما تتحقق قيمة واحدة أو كل القيم.

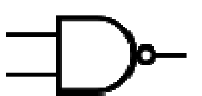
بوابات XOR:



Input		Output
A	B	A xor B
0	0	0
0	1	1
1	0	1
1	1	0

البوابة "أكسور": تصبح القيمة واحد عندما لا تتشابه كل القيم.

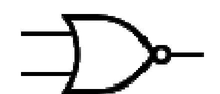
بوابات NAND:



Input		Output
A	B	A nand B
0	0	1
0	1	1
1	0	1
1	1	0

البوابة "ناند": تصبح القيمة واحد في جميع الأحوال ماعدا في حالة الوحايد.

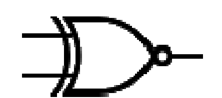
بوابات NOR:



Input		Output
A	B	A nor B
0	0	1
0	1	0
1	0	0
1	1	0

البوابة "نور": تصبح القيمة صفر فى جميع الأحوال ماعدا فى حالة الأصفار.

بوابات XNOR:



Input		Output
A	B	A xnor B
0	0	1
0	1	0
1	0	0
1	1	1

البوابة "أكسنور": تصبح القيمة واحد فى حالة تشابه القيم فقط أى فى حالة الوحايد والأصفار.

لا توجد بوابات مباشرة بأسم NAND, أو NOR, أو XOR لأنه يمكن التوصل إليها باستخدام البوابات AND أو OR أو XOR مع البوابة INV.

الشرح:

الشكل	الشرح	الأسم	م
INV_B: EN ENO; ????-IN OUT-????	يقوم INV_B بعكس محتويات الـ Bit من الـ Byte IN إلى الـ Byte OUT.	INV_B	١
INV_W: EN ENO; ????-IN OUT-????	يقوم INV_W بعكس محتويات الـ Bit من الـ IN Word إلى الـ Word OUT.	INV_W	٢
INV_DW: EN ENO; ????-IN OUT-????	يقوم INV_DW بعكس محتويات الـ Bit من الـ IN Dword إلى الـ Dword OUT.	INV_DW	٣
WAND_B: EN ENO; ????-IN1 OUT-????; ????-IN2	يقوم AND_B بتنفيذ عملية "و" بين محتويات الـ Bits الخاصة بالـ Byte	WAND_B	٤

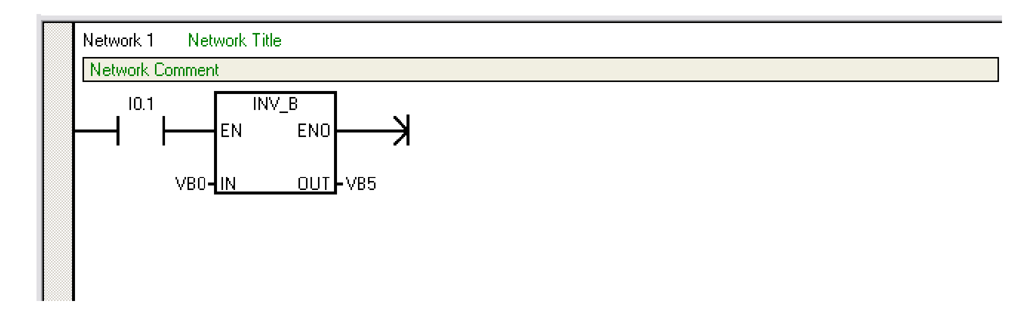
		IN1 و محتويات ال Bits الخاصة بال Byte IN2.	
٥	WAND_W	يقوم AND_W بتنفيذ عملية "و" بين محتويات ال Bits الخاصة بال Word IN1 و محتويات ال Bits الخاصة بال Word IN2.	WAND_W EN ENO ????-IN1 OUT-???? ????-IN2
٦	WAND_DW	يقوم AND_DW بتنفيذ عملية "و" بين محتويات ال Bits الخاصة بال Dword IN1 و محتويات ال Bits الخاصة بال Dword IN2.	WAND_DW EN ENO ????-IN1 OUT-???? ????-IN2
٧	WOR_B	يقوم WOR_B بتنفيذ عملية "أو" بين محتويات ال Bits الخاصة بال Byte IN1 و محتويات ال Bits الخاصة بال Byte IN2.	WOR_B EN ENO ????-IN1 OUT-???? ????-IN2

٨	WOR_W	يقوم WOR_W بتنفيذ عملية "أو" بين محتويات الـ Bits الخاصة بالـ Word IN1 و محتويات الـ Bits الخاصة بالـ Word IN2.	WOR_W EN ENO ????-IN1 OUT-???? ????-IN2
٩	WOR_DW	يقوم WOR_DW بتنفيذ عملية "أو" بين محتويات الـ Bits الخاصة بالـ Dword IN1 و محتويات الـ Bits الخاصة بالـ Dword IN2.	WOR_DW EN ENO ????-IN1 OUT-???? ????-IN2
١٠	WXOR_B	يقوم WXOR_B بتنفيذ عملية "أكسور" بين محتويات الـ Bits الخاصة بالـ Byte IN1 و محتويات الـ Bits الخاصة بالـ Byte IN2.	WXOR_B EN ENO ????-IN1 OUT-???? ????-IN2
١١	WXOR_W	يقوم WXOR_W بتنفيذ عملية "أكسور" بين محتويات الـ Bits الخاصة بالـ Word IN1 و	WXOR_W EN ENO ????-IN1 OUT-???? ????-IN2

		محتويات ال Bits الخاصة بال Word IN2.	
١٢	WXOR_DW	يقوم WXOR_DW بتنفيذ عملية "أكسور" بين محتويات ال Bits الخاصة بال IN1 Dword و محتويات ال Bits الخاصة بال Dword IN2.	WXOR_DW EN ENO ????-IN1 OUT-???? ????-IN2

مثال:

تمرين باستخدام INV_B.



VB0

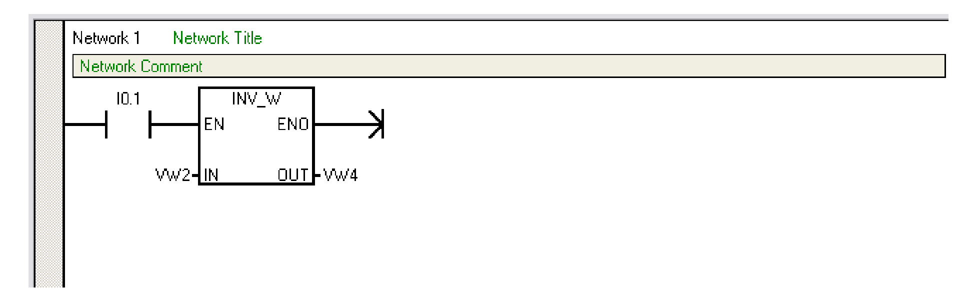
1	0	1	0	0	0	1	1

VB5

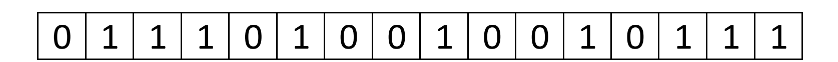
0	1	0	1	1	1	0	0

مثال:

تمرين باستخدام INV_W.



VW2



0	1	1	1	0	1	0	0	1	0	0	1	0	1	1	1

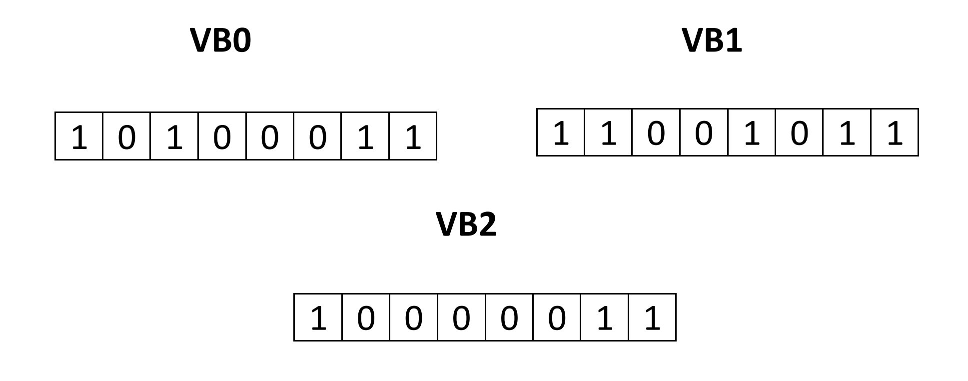
VW4

1	0	0	0	1	0	1	1	0	1	1	0	1	0	0	0

مثال:

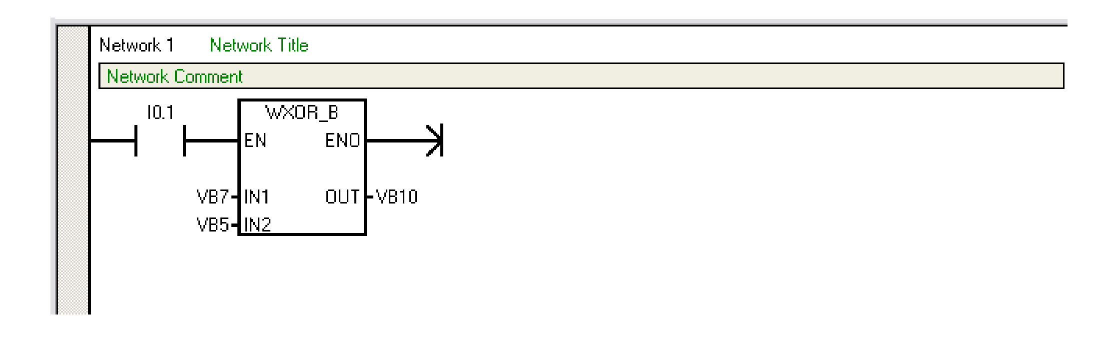
تمرين باستخدام WAND_B.

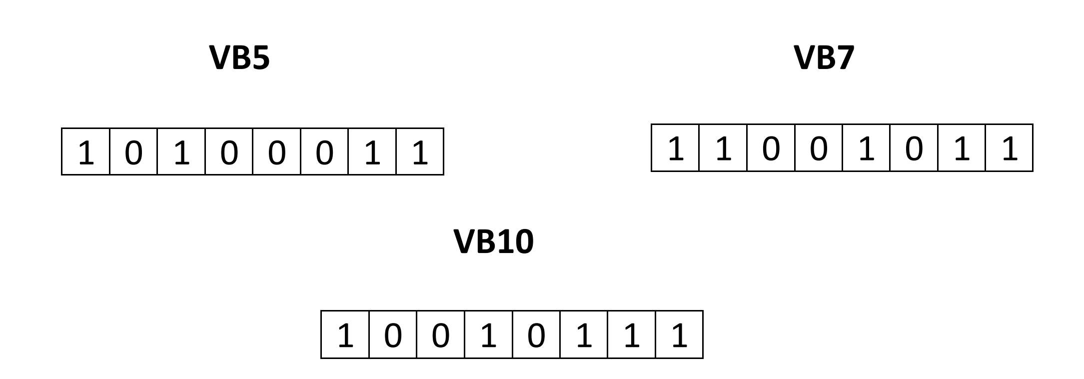
Network 1 Network Title
Network Comment
I0.1
WAND_B
EN ENO
VB0 IN1 OUT VB2
VB1 IN2



مثال:

تمرين باستخدام WXOR_B.

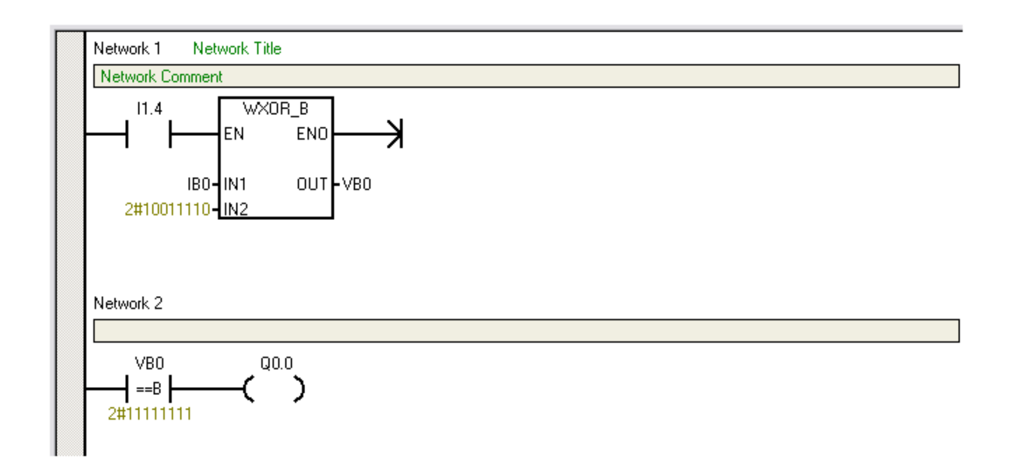




مثال عملى:

قم بتنفيذ دائرة تحكم منطقية لمحرك يعمل بشرط أن تكون حالة المفاتيح كالأتى:

I0.7	I0.6	I0.5	I0.4	I0.3	I0.2	I0.1	I0.0
1	0	0	1	1	1	1	0



:Network1

يتم تطبيق العملية "XOR" بين IB0 و 2#1001-1110 و وضع النتيجة فى VB0

:Network2

تعمل Q0.0 أذا كانت قيمة VB0 هى 2#1111-1111.

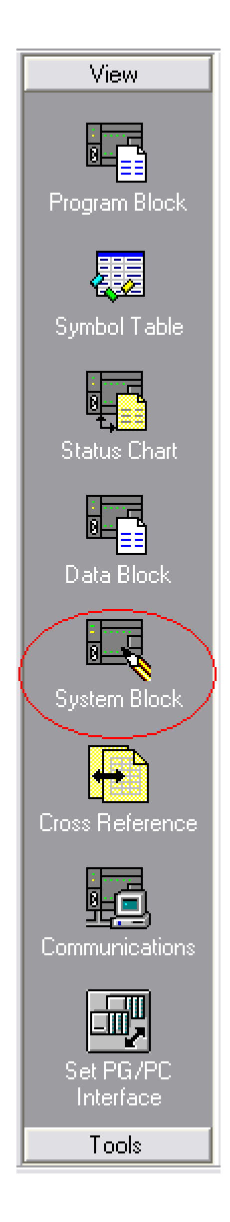
الباب الثامن

النظم العملية

- شرح النظـــــــــــــــــم العملية.
- صفحـــة ال Communication Ports.
- صفحـــــــــــة Retentive Ranges.
- صفحـــــــــــــــــة Password.
- صفحة Output Tables DIGITAL.
- صفحــــة Input Filters DIGITAL.
- صفحـــــــــة ال Pulse Chatch Bits.
- صفحـــــــــة ال Background Time.
- صفحــــــــــة ال Configure LED.
- صفحــــــــة ال Increase Memory.
- المفاتيح المستخدمة فى صفحة النظـم العمليــة.
- الأخــــــــــطاء الممكن التعرض لها.

صفحة النظم العملية:

تستخدم صفحة "النظم العملية" الـ System Block لتحديد بعض المتغيرات الخاصة بالبرنامج والتى يمكن لها أن تغير فى طبيعة عمل البرنامج من حيث طريقة التشغيل.



طريقة استخدام صفحة "النظم العملية":

- يتم عمل البرنامج أولاً ولكن قبل تحميل البرنامج يتم تحديد المتغيرات بواسطة صفحة " النظم العملية " وبعد ذالك يتم تحميل البرنامج.

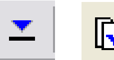
ملاحظة:

فى حاله رسم البرنامج و تحميله ثم تعديل بعض المتغيرات باستخدام صفحة " النظم العملية " دون تحميل البرنامج مرة أخرى فأن كل المتغيرات تعتبر غير فعالة ولذالك سيلحظ دائماً ظهور رسالة فى جميع الصفح الفرعية داخل صفحة "النظم العملية" وهى:

" Configuration parameters must be downloaded before they take effect"

أى أنه "لن يحدث أى تغير قبل تنفيذ عملية التحميل"

المفاتيح المستخدمة فى صفحة الرموز.



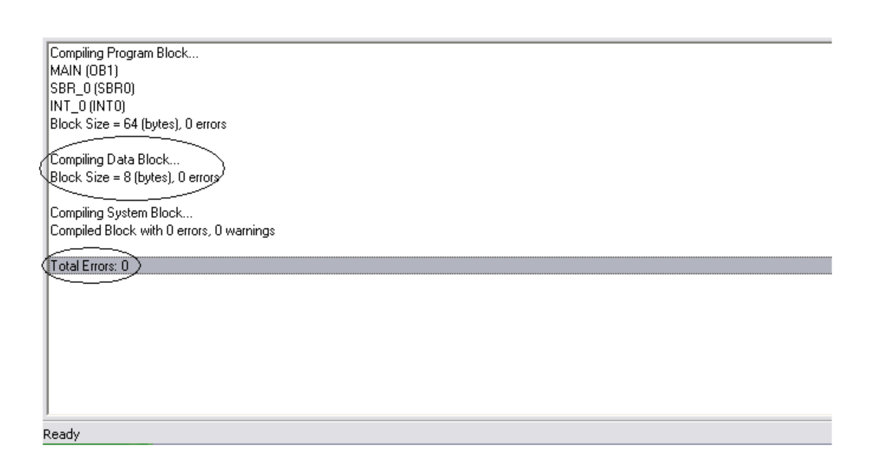
:Download

بالضغط على هذا المفتاح يتم تحميل البرنامج كما سبق و شرحنا و تصبح جميع المتغيرات المحددة بواسطة صفحة "النظم العملية" متاحة.

Compile all

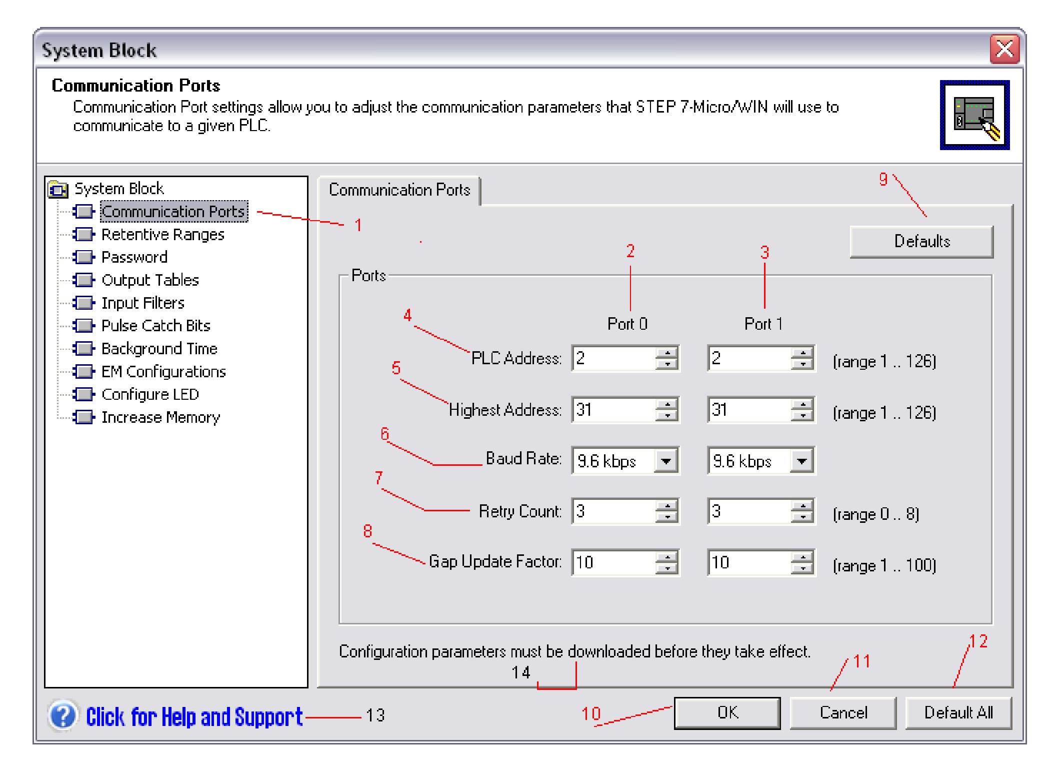
بالضغط على هذا المفتاح تظهر عدد الأعطال الخاصة بالبرنامج لكى يتم تجنبها قبل تطبيق أمر التحميل.

بالضغط على compile all تظهر هذه الرسالة التى توضح الأخطاء إذا وجدت:



١- صفحة Communication Ports.

هى صفحة لتحديد المتغيرات الخاصة بالتواصل بين وحدة الـ PLC و جهاز التحكم أى أن كان نوعه.



١- صفحة الـ Communication Ports.

٢- المقصود بكلمة Port0 هى البيانات الخاصة بوحدة الـ PLC.

٣- المقصود بكلمة Port1 هى البيانات الخاصة بالوحدة الإضافية.

٤- المقصود بكلمة PLC Address هو العنوان الخاص بوحدة الـ PLC وغالباً ما يكون أثنان.

٥- المقصود بكلمة Highest Address هو أقصى أسم لوحدات الـ PLC يمكن للحاسب الألى التواصل معهم.

٦- المقصود بكلمة Baud Rate هى سرعة تبادل البيانات بين الحاسب الألى و وحدة الـ PLC.

٧- المقصود بكلمة Retry Count هى عدد محاولات أعادة التواصل بين الحاسب الألى و وحدة الـ PLC.

٨- المقصود بكلمة Gap Update Factor هى الفراغات بين وحدات الـ PLC ويفضل بأن يكون عدد ليس بكبير.

٩- المقصود بكلمة Default هى أعادة المتغيرات الخاصة بصفحة Communication Ports فقط إلى طبيعتها.

١٠- المقصود بكلمة Ok هو تفعيل جميع المتغيرات التى نفذت و لكن بعد تحميل البرنامج.

١١- المقصود بكلمة Cancel هو إلغاء التعديلات و غلق الصفحة.

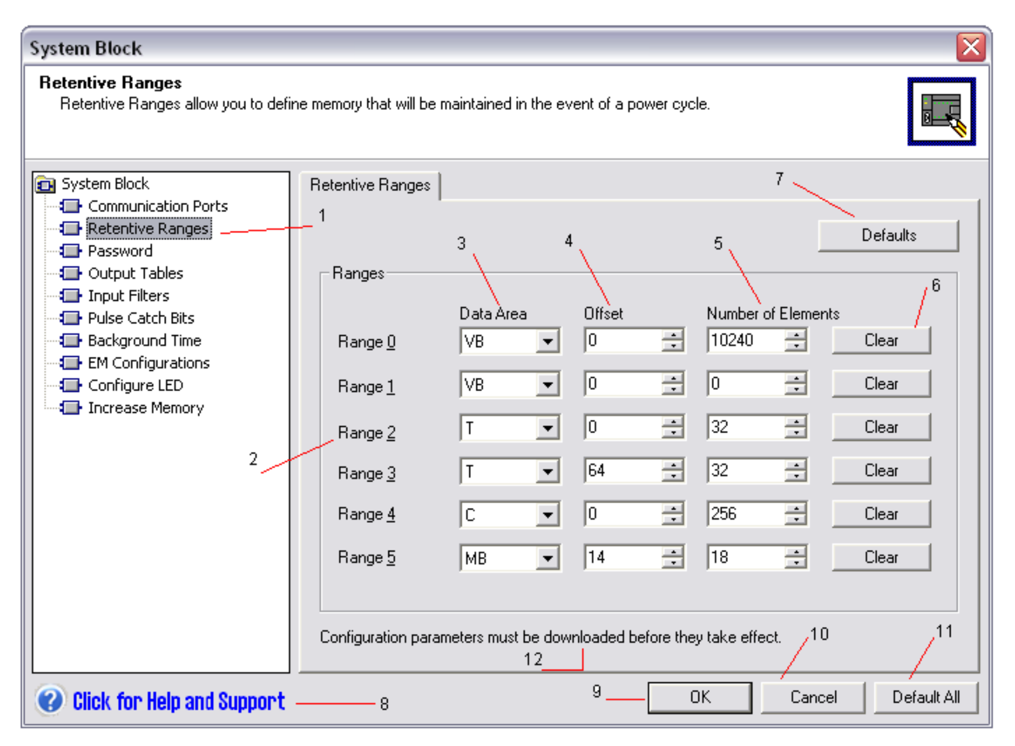
١٢- المقصود بكلمة Default All هى أعادة المتغيرات الخاصة بجميع صفح System Block إلى طبيعتها.

١٣- المقصود بكلمة Help هى صفحة للتوضيح و المساعدة خاصة بصفحة Communication Ports فقط.

١٤- لتفعيل أى من المتغيرات التى سوف يتم تطبيقها بواسطة صفحة Communication Ports يجب تحميل البرنامج.

٢- صفحة Retentive Ranges.

تستخدم هذه الصفحة لتطبيق أمر Retentive على ريليهات أو مؤقتات زمنية أو عدادات لكى يحتفظوا بقيمتهم.



١- صفحة الـ Retentive Ranges.

٢- المقصود بكلمة Range هى المجموعات التى سوف يتم التعامل معها.

٣- المقصود بكلمة Data Area هى القائمة التى تحتوى على العناوين المستخدمة فى صفحة الـ Retentive Ranges.

٤- المقصود بكلمة Offset هو أسم العنوان الذى سوف يتم تطبيق مبدأ الـ Retentive عليه.

٥- المقصود بكلمة Number of Elements هى عدد العناوين التى سوف يتم تطبيق مبدأ الـ Retentive عليه بدايتاً من العنوان المدون فى الـ Offset.

٦- المقصود بكلمة Clear هو محو جميع التعديلات الخاصة بالصف المجاور لها.

٧- المقصود بكلمة Default هى أعادة المتغيرات الخاصة بصفحة Retentive Ranges فقط إلى طبيعتها.

٨- المقصود بكلمة Help هى صفحة للتوضيح و المساعدة خاصة بصفحة Retentive Ranges فقط.

٩- المقصود بكلمة Ok هو تفعيل جميع المتغيرات التى نفذت و لكن بعد تحميل البرنامج.

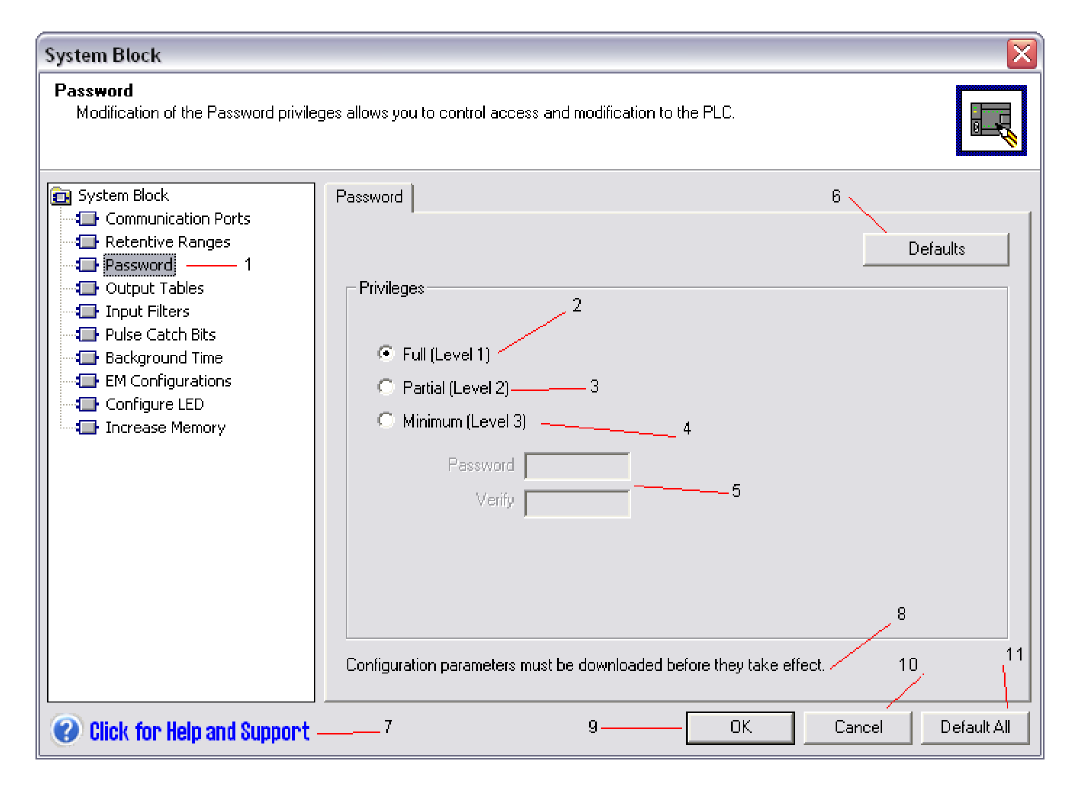
١٠- المقصود بكلمة Cancel هو إلغاء التعديلات و غلق الصفحة.

١١- المقصود بكلمة Default All هى أعادة المتغيرات الخاصة بجميع صفح Retentive Ranges إلى طبيعتها.

١٢- لتفعيل أى من المتغيرات التى سوف يتم تطبيقها بواسطة صفحة Retentive Ranges يجب تحميل البرنامج.

٣- صفحة Password.

تستخدم هذه الصفحة لوضع كلمة مرور لتحديد بعض الأعمال.



١- صفحة ال Password.

٢- المقصود بكلمة Full level هو أقل مستوى من مستويات كلمات المرور حيث أنه لا تأثير لكلمة المرور الخاصة بهذا المستوى.

٣- المقصود بكلمة Partial level هو ثانى مستوى من مستويات كلمات المرور حيث أنه لا يمكن تحميل البرنامج أو تطبيق أمر Force أثناء تفعيل كلمة المرور الخاصة بهذا المستوى.

٤- المقصود بكلمة Minimum level هو أقوى مستوى من مستويات كلمات المرور حيث أنه لا يمكن تحميل البرنامج download أو تطبيق أمر Force أو أمر Upload أثناء تفعيل كلمة المرور الخاصة بهذا المستوى.

٥- المقصود بهذا الفراغ هو المكان الذى يتم كتابة كلمة المرور بداخله مرتين.

٦- المقصود بكلمة Default هى أعادة المتغيرات الخاصة بصفحة Password فقط إلى طبيعتها.

٧- المقصود بكلمة Help هي صفحة للتوضيح و المساعدة خاصة بصفحة Password فقط.

٨- لتفعيل أي من المتغيرات التي سوف يتم تطبيقها بواسطة صفحة Password يجب تحميل البرنامج.

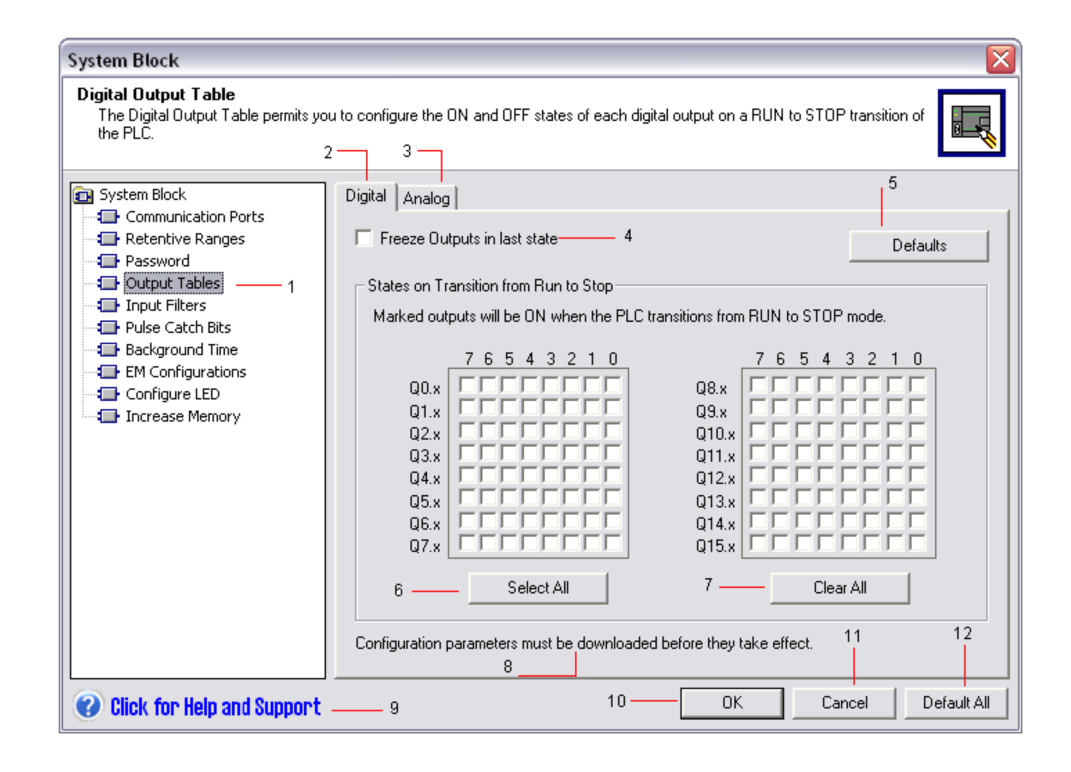
٩- المقصود بكلمة Ok هو تفعيل جميع المتغيرات التي نفذت و لكن بعد تحميل البرنامج.

١٠- المقصود بكلمة Cancel هو إلغاء التعديلات و غلق الصفحة.

١١- المقصود بكلمة Default All هي أعادة المتغيرات الخاصة بجميع صفح System block إلى طبيعتها.

٤- صفحة Output Tables DIGITAL.

تستخدم هذه الصفحة لإختيار المخرجات Digital المراد أن تحتفظ بأخر حالة لها عند تحويل وضع وحدة الـ PLC من عمل إلى إيقاف.



١- صفحة الـ Output Tables DIGITAL.

٢- المقصود بكلمة Digital هى صفحة خاصة بالخرج الرقمى.

٣- المقصود بكلمة Analog هى صفحة خاصة بالخرج التناظرى.

٤- المقصود بكلمة Freeze Output in last state هو تثبيت الخرج على أخر حاله له.

٥- المقصود بكلمة Default هى أعادة المتغيرات الخاصة بصفحة Output Tables DIGITAL فقط إلى طبيعتها.

٦- المقصود بكلمة Select All هو إختيار جميع المخرجات لكى يطبق عليها أمر التثبيت Freeze.

٧- المقصود بكلمة Clear All هو عدم إختيار جميع المخرجات لكى لا يطبق عليها أمر التثبيت Freeze.

٨- لتفعيل أى من المتغيرات التى سوف يتم تطبيقها بواسطة صفحة Output Tables DIGITAL يجب تحميل البرنامج.

٩- المقصود بكلمة Help هى صفحة للتوضيح و المساعدة خاصة بصفحة Output Tables DIGITAL فقط.

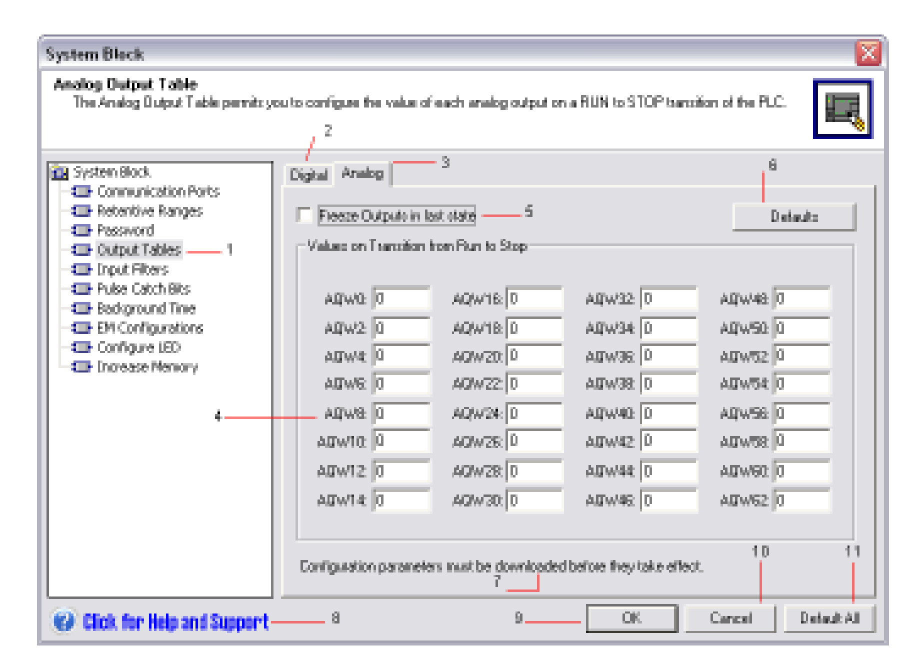
١٠- المقصود بكلمة Ok هو تفعيل جميع المتغيرات التى نفذت و لكن بعد تحميل البرنامج.

١١- المقصود بكلمة Cancel هو إلغاء التعديلات و غلق الصفحة.

١٢- المقصود بكلمة Default All هى أعادة المتغيرات الخاصة بجميع صفحات System block إلى طبيعتها.

٥- صفحة Output Tables ANALOG.

تستخدم هذه الصفحة لأختيار المخرجات التناظرية المراد أن تحتفظ بأخر حالة "قيمة" لها عند تحويل وضع وحدة الـ PLC من عمل إلى إيقاف.



١- صفحة الـ Output Tables ANALOG.

٢- المقصود بكلمة Digital هى صفحة خاصة بالخرج الرقمى.

٣- المقصود بكلمة Analog هى صفحة خاصة بالخرج التناظرى.

٤- المقصود بكلمة AQW هو أسم الخرج التناظرى.

٥- المقصود بكلمة Freeze Output in last state هو تثبيت الخرج على أخر قيمة له.

٦- المقصود بكلمة Default هى أعادة المتغيرات الخاصة بصفحة Output Tables Analog فقط إلى طبيعتها.

٧- لتفعيل أى من المتغيرات التى سوف يتم تطبيقها بواسطة صفحة Output Tables ANALOG يجب تحميل البرنامج.

٨- المقصود بكلمة Help هى صفحة للتوضيح و المساعدة خاصة بصفحة Output Tables ANALOG فقط.

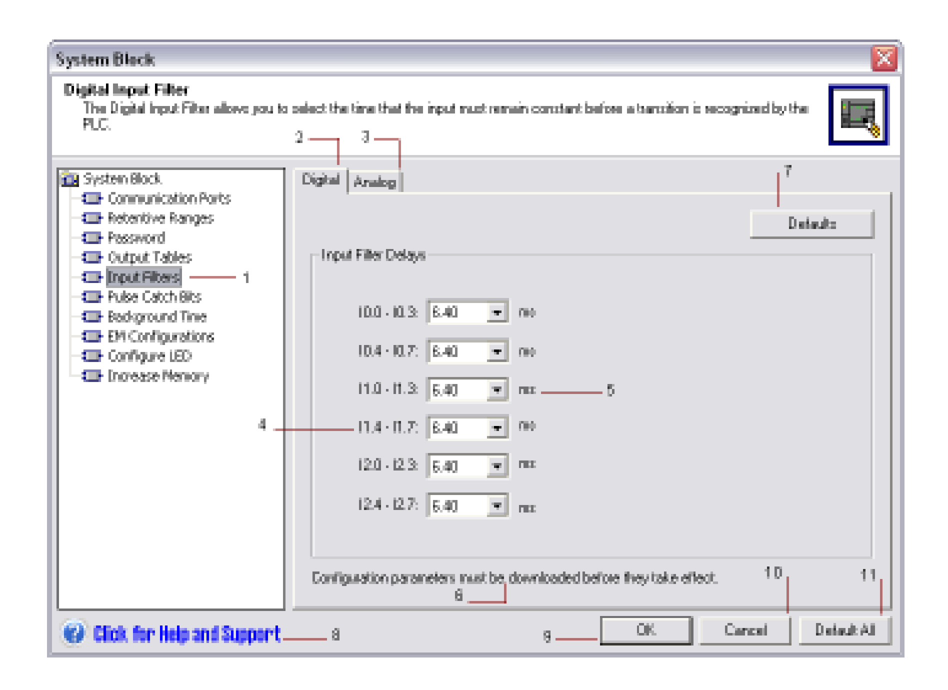
٩- المقصود بكلمة Ok هو تفعيل جميع المتغيرات التى نفذت و لكن بعد تحميل البرنامج.

١٠- المقصود بكلمة Cancel هو إلغاء التعديلات و غلق الصفحة.

١١- المقصود بكلمة Default All هى أعادة المتغيرات الخاصة بجميع صفح System block إلى طبيعتها.

٦- صفحة Input Filters DIGITAL.

تستخدم هذه الصفحة لاختيار زمن الإشارة الخاصة بالمداخلات Digital.



١- صفحة ال Input Filters DIGITAL.

٢- المقصود بكلمة Digital هى صفحة خاصة بالخرج الرقمى.

٣- المقصود بكلمة Analog هى صفحة خاصة بالخرج التناظرى.

٤– المقصود به مجموعة من المدخلات التى سوف يطبق عليها أمر **Input Filter**.

٥– المقصود به الزمن المراد تطبيقه.

٦– لتفعيل أى من المتغيرات التى سوف يتم تطبيقها بواسطة صفحة Input Filters DIGITAL يجب تحميل البرنامج.

٧– المقصود بكلمة Default هى أعادة المتغيرات الخاصة بصفحة Input Filters DIGITAL فقط إلى طبيعتها.

٨– المقصود بكلمة Help هى صفحة للتوضيح و المساعدة خاصة بصفحة Input Filters DIGITAL فقط.

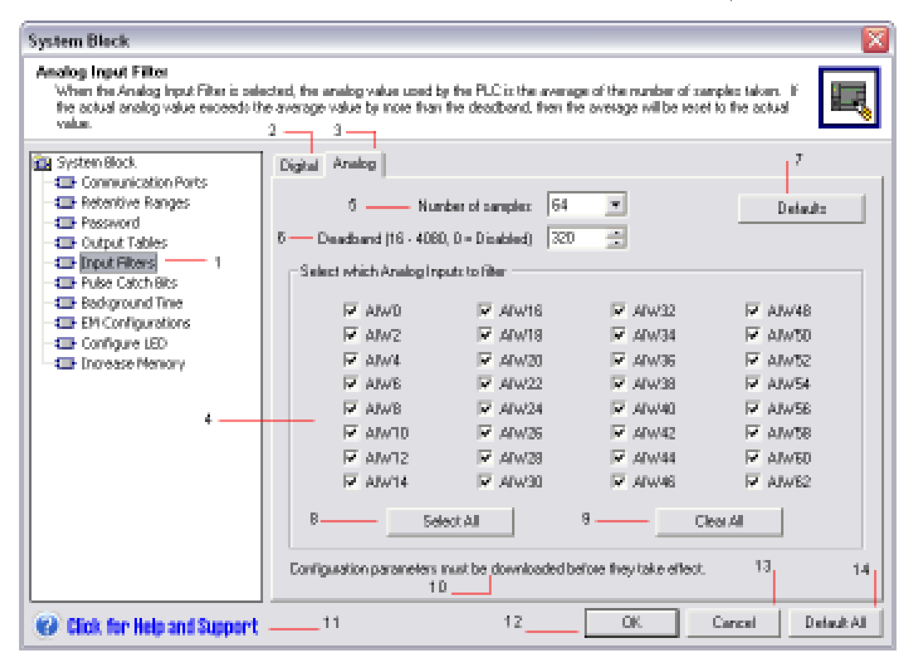
٩– المقصود بكلمة Ok هو تفعيل جميع المتغيرات التى نفذت و لكن بعد تحميل البرنامج.

١٠– المقصود بكلمة Cancel هو إلغاء التعديلات و غلق الصفحة.

١١– المقصود بكلمة Default All هى أعادة المتغيرات الخاصة بجميع صفح System block إلى طبيعتها.

٧- صفحة Input Filters ANALOG.

تستخدم هذه الصفحة للتحديد المسبق لقيمة الإشارة الخاصة بالمدخلات التناظرية.



١- صفحة الـ Input Filters ANALOG.

٢- المقصود بكلمة Digital هي صفحة خاصة بالخرج الرقمي.

٣- المقصود بكلمة Analog هي صفحة خاصة بالخرج .

٤- المقصود به مجموعة من المدخلات التي سوف يطبق عليها أمر Input Filter.

٥- المقصود بكلمة Number of samples حيث يمكن تحديد قيمة مسبقة للمدخلات التناظرية.

٦- المقصود بكلمة Deadband حيث يتم تحديد التغيير الذى يسمح بقبوله من المدخلات التناظرية.

٧- المقصود بكلمة Default هي أعادة المتغيرات الخاصة بصفحة Input Filters ANALOG فقط إلى طبيعتها.

٨- المقصود بكلمة Select All هو اختيار جميع المدخلات لكى يتم تفعيلها.

٩- المقصود بكلمة Clear All هو عدم اختيار أى من المدخلات لكى لا يطبق عليها أمر Filter.

١٠- لتفعيل أى من المتغيرات التى سوف يتم تطبيقها بواسطة صفحة Input Filters ANALOG يجب تحميل البرنامج.

١١- المقصود بكلمة Help هى صفحة للتوضيح و المساعدة خاصة بصفحة Input Filters ANALOG فقط.

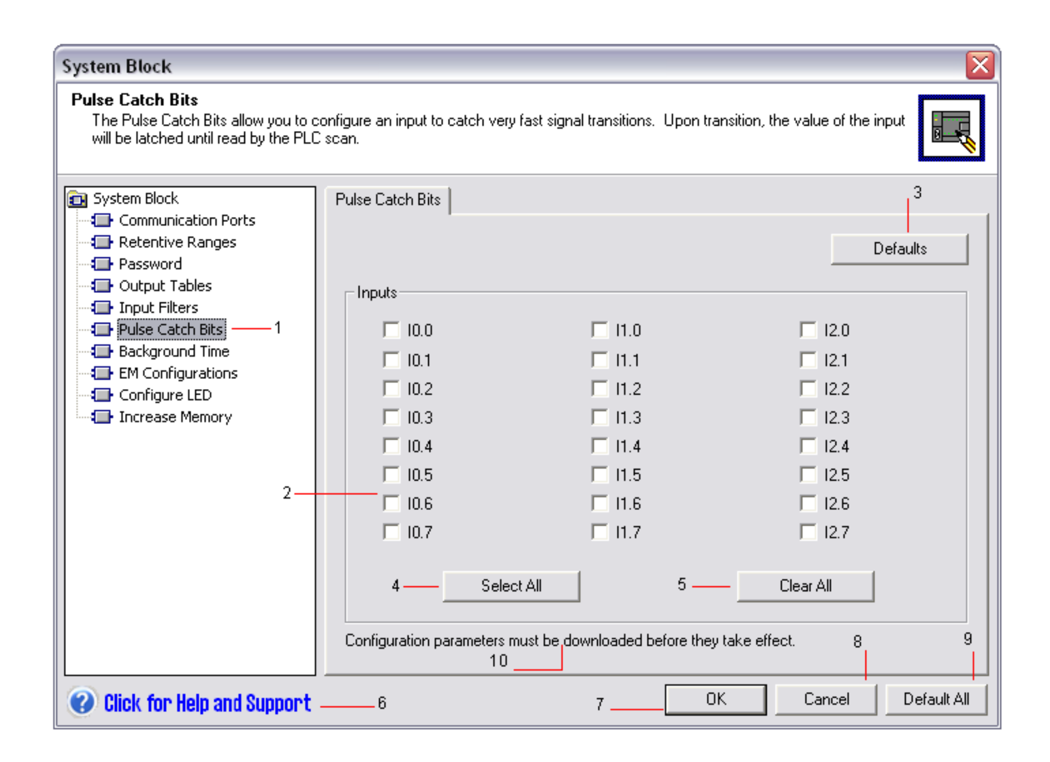
١٢- المقصود بكلمة Ok هو تفعيل جميع المتغيرات التى نفذت و لكن بعد تحميل البرنامج.

١٣- المقصود بكلمة Cancel هو إلغاء التعديلات و غلق الصفحة.

١٤- المقصود بكلمة Default All هى أعادة المتغيرات الخاصة بجميع صفح Input Filters ANALOG إلى طبيعتها.

٨- صفحة Pulse catch Bits.

تستخدم هذه الصفحة للاحتفاظ بزمن الإشارة الخاصة بالمدخلات حتى يستكمل دورة كاملة إضافية.



١- صفحة الـ Pulse Catch Bits.

٢- حيث يتم اختيار المدخلات المراد التعامل معها.

٣- المقصود بكلمة Default هى أعادة المتغيرات الخاصة بصفحة Pulse Catch Bits فقط إلى طبيعتها.

٤- المقصود بكلمة Select All هو اختيار جميع المدخلات لكى يتم تفعيلها.

٥- المقصود بكلمة Clear All هو عدم اختيار أى من المدخلات لكى لا يطبق عليها أمر Catch.

٦- المقصود بكلمة Help هى صفحة للتوضيح و المساعدة خاصة بصفحة Pulse Catch Bits فقط.

٧- المقصود بكلمة Ok هو تفعيل جميع المتغيرات التى نفذت و لكن بعد تحميل البرنامج.

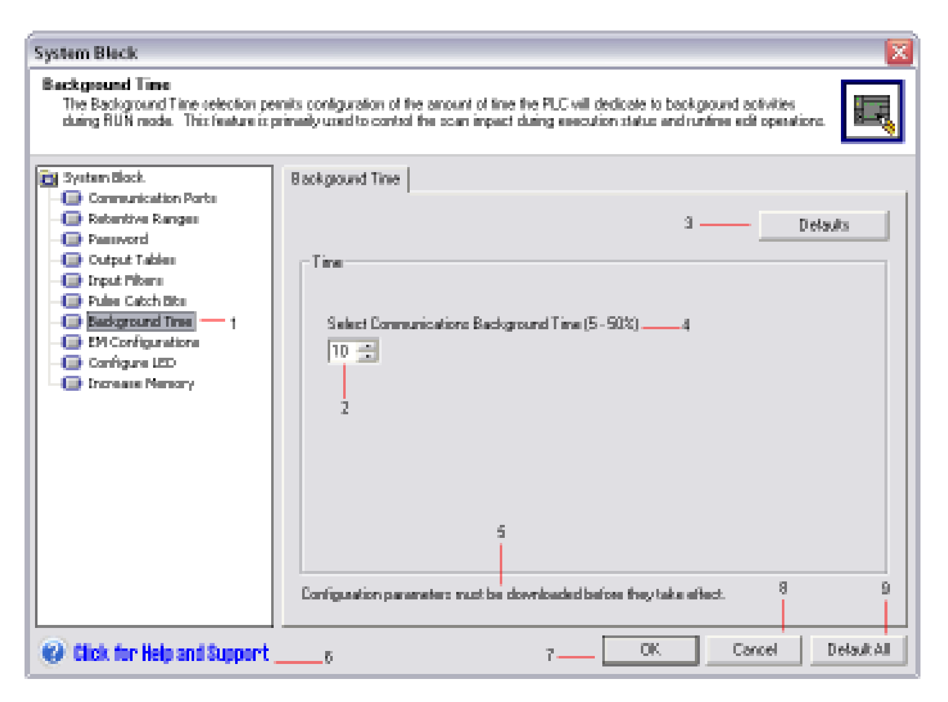
٨- المقصود بكلمة Cancel هو إلغاء التعديلات و غلق الصفحة.

٩- المقصود بكلمة Default All هى أعادة المتغيرات الخاصة بجميع صفح System block إلى طبيعتها.

١٠- لتفعيل أى من المتغيرات التى سوف يتم تطبيقها بواسطة صفحة Pulse Catch Bits يجب تحميل البرنامج.

٩- صفحة Background Time.

تستخدم هذه الصفحة للتحكم فى سرعة الأمر التوضيحى program status.



١ – صفحة الـ Background Time.

٢ – المقصود به هو النسبة المئوية للتحكم بالزمن الفعلى.

٣- المقصود بكلمة Default هى أعادة المتغيرات الخاصة بصفحة Background Time فقط إلى طبيعتها.

٤ – يمكن التحكم بالزمن حتى أنه يمكن أبطاء الزمن الحقيقى من ٥% إلى ٥٠%.

٥- لتفعيل أى من المتغيرات التى سوف يتم تطبيقها بواسطة صفحة Background Time يجب تحميل البرنامج.

٦- المقصود بكلمة Help هى صفحة للتوضيح و المساعدة خاصة بصفحة Background Time فقط.

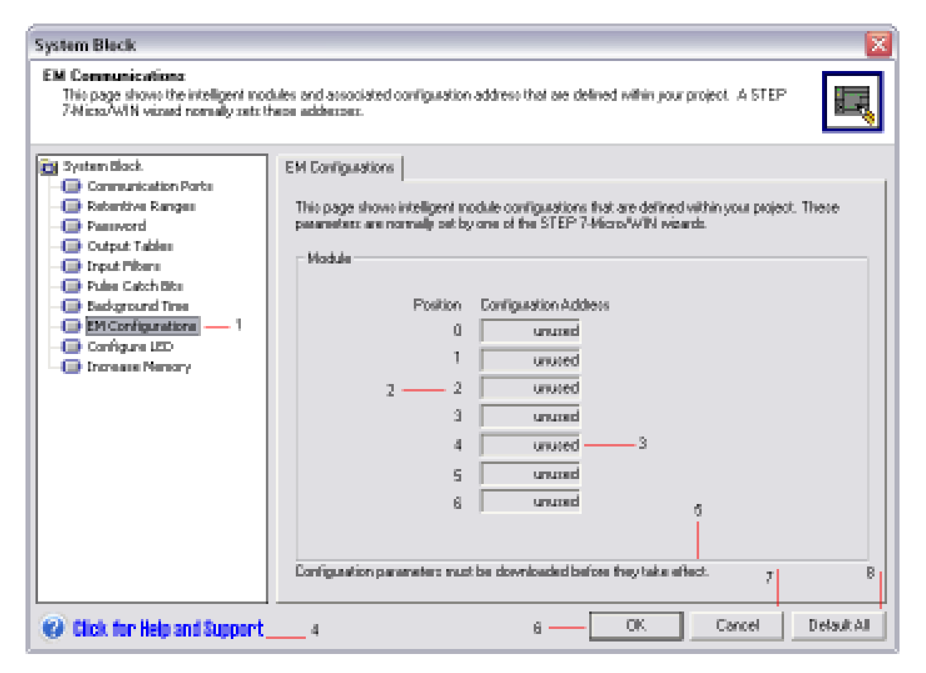
٧- المقصود بكلمة Ok هو تفعيل جميع المتغيرات التى نفذت و لكن بعد تحميل البرنامج.

٨- المقصود بكلمة Cancel هو إلغاء التعديلات و غلق الصفحة.

٩- المقصود بكلمة Default All هى أعادة المتغيرات الخاصة بجميع صفح System block إلى طبيعتها.

١٠- صفحة EM Configurations.

تستخدم هذه الصفحة للتحكم فى سرعة الأمر التوضيحى program status.



١- صفحة ال EM Configurations.

٢- المقصود به هو تسلسل وحدة المدخلات أو المخرجات الإضافية.

٣- المقصود به هو توضيح أذا كان وحدة المدخلات أو المخرجات الإضافية مستخدمة أم غير مستخدمة.

٤- لتفعيل أى من المتغيرات التى سوف يتم تطبيقها بواسطة صفحة EM Configurations يجب تحميل البرنامج.

٥- المقصود بكلمة Help هى صفحة للتوضيح و المساعدة خاصة بصفحة EM Configurations فقط.

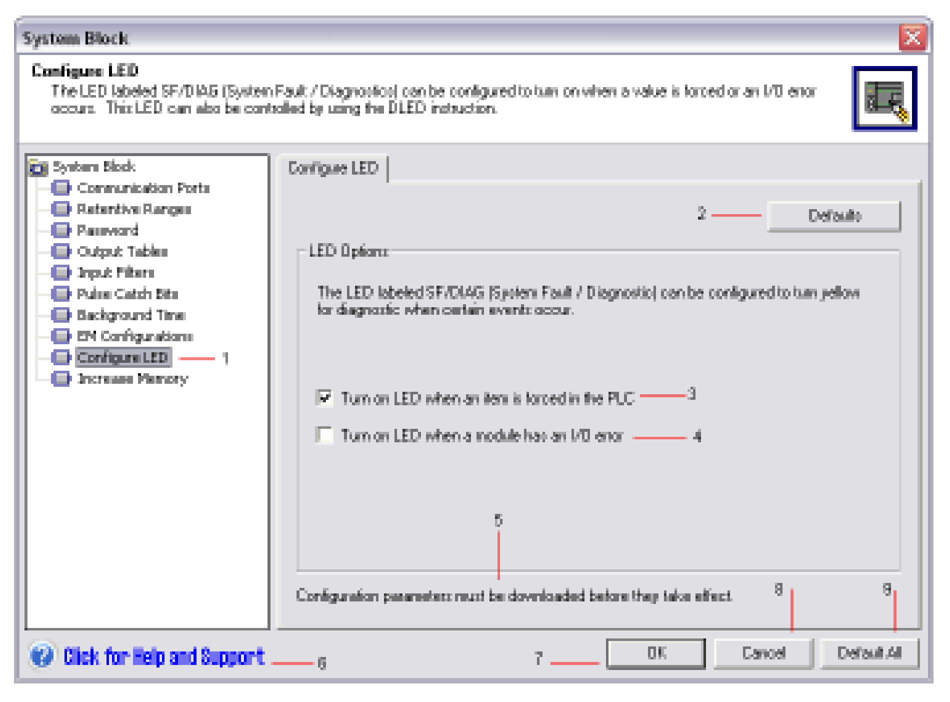
٦- المقصود بكلمة Ok هو تفعيل جميع المتغيرات التى نفذت و لكن بعد تحميل البرنامج.

٧- المقصود بكلمة Cancel هو إلغاء التعديلات و غلق الصفحة.

٨- المقصود بكلمة Default All هى أعادة المتغيرات الخاصة بجميع صفح System block إلى طبيعتها.

١١- صفحة Configure LED.

تستخدم هذه الصفحة للتحكم فى سرعة الأمر التوضيحى program status.



١- صفحة الـ Configure LED.

٢- المقصود بكلمة Default هى أعادة المتغيرات الخاصة بصفحة Configure LED فقط إلى طبيعتها.

٣- المقصود به هو أنه سوف تضيء لمبة الأعطال فى حالة استخدام أمر Force.

٤- المقصود به هو أنه سوف تضيء لمبة الأعطال فى حالة وجود أعطال خاصة بالمداخلات أو بالمخرجات.

٥- لتفعيل أى من المتغيرات التى سوف يتم تطبيقها بواسطة صفحة Configure LED يجب تحميل البرنامج.

٦- المقصود بكلمة Help هى صفحة للتوضيح و المساعدة خاصة بصفحة Configure LED فقط.

٧- المقصود بكلمة Ok هو تفعيل جميع المتغيرات التى نفذت و لكن بعد تحميل البرنامج.

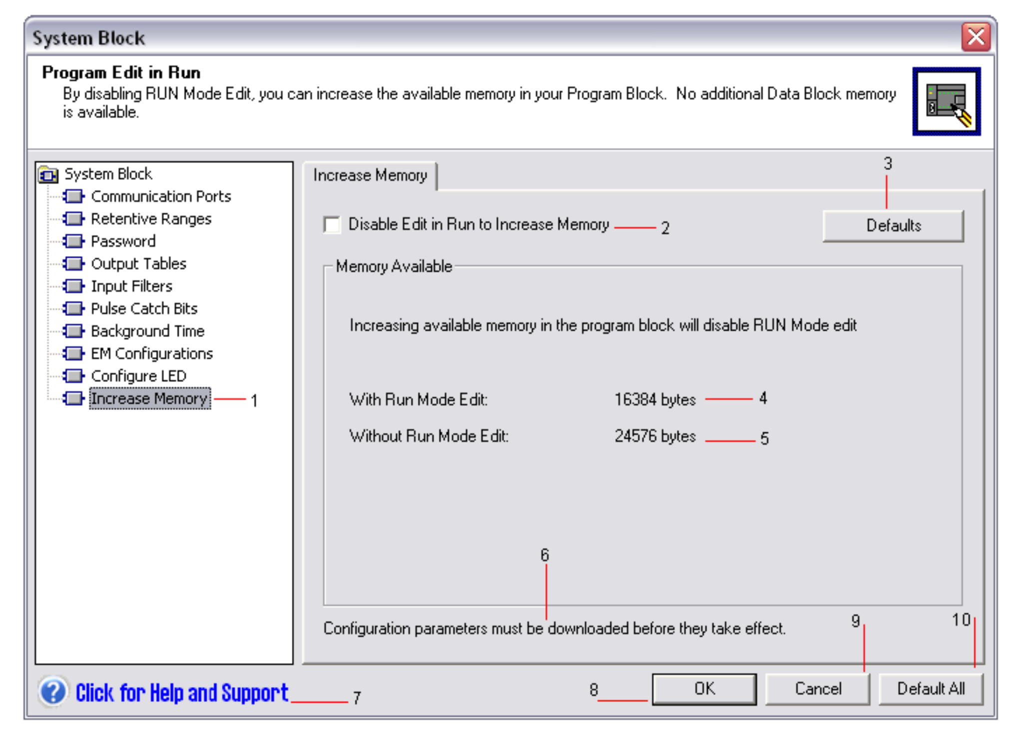
٨- المقصود بكلمة Cancel هو إلغاء التعديلات و غلق الصفحة.

٩- المقصود بكلمة Default All هى أعادة المتغيرات الخاصة بجميع صفح System block إلى طبيعتها.

١٢- صفحة Increase Memory.

تستخدم هذه الصفحة للتحكم فى حجم الذاكرة سواء بالتعديل فى البرنامج أثناء تشغيل أو أثناء إيقاف الـ PLC:

- حيث أنة فى حالة تعديل البرنامج أثناء العمل فهذا يعنى أنه يجب على الذاكرة احتواء البرنامج القديم حتى يتم تفعيل البرنامج الجديد.
- حيث أنة فى حالة تعديل البرنامج أثناء التوقف فهذا يعنى أنه يجب على الذاكرة محو البرنامج القديم ثم تحميل البرنامج الجديد.



١– صفحة الـ Increase Memory.

٢– المقصود به هو عدم التعديل فى البرنامج أثناء عمل وحدة الـ PLC.

٣– المقصود بكلك Default هى أعادة المتغيرات الخاصة بصفحة Increase Memory فقط إلى طبيعتها.

٤– المقصود به هو أن التعديل فى البرنامج أثناء عمل وحدة الـ PLC يوفر فى الذاكرة المستخدمة فى البرمجة.

٥– المقصود به هو أن التعديل فى البرنامج أثناء وقوف وحدة الـ PLC لا يوفر فى الذاكرة المستخدمة فى البرمجة.

٦– لتفعيل أى من المتغيرات التى سوف يتم تطبيقها بواسطة صفحة Increase Memory يجب تحميل البرنامج.

٧– المقصود بكلك Help هى صفحة للتوضيح و المساعدة خاصة بصفحة Increase Memory فقط.

٨- المقصود بكلمة Ok هو تفعيل جميع المتغيرات التى نفذت و لكن بعد تحميل البرنامج.

٩- المقصود بكلمة Cancel هو إلغاء التعديلات و غلق الصفحة.

١٠- المقصود بكلمة Default All هى أعادة المتغيرات الخاصة بجميع صفح System block إلى طبيعتها.

الباب التاسع

الريليهات الخاصة

- شرح الريليهـــــــــــــــــــــات الخاصة.
- أنواع الريليهـــــــــــــــــــات الخاصة.
- صفحة الريليهـــــــــــــــات الخاصة.
- تعليقات الريليهـــــــــــــــات الخاصة.
- رموز الريليهـــــــــــــــــــات الخاصة.
- المفاتيح المستخدمة مع الريليهـات الخاصة.
- تمـــــــــــــــــــــارين عملية للتوضيح.

الريليهات الخاصة:

تعتبر الريليهات الخاصة هى عبار عن ذاكرة خاصة داخل وحدة ال PLC تعمل بطرق مختلفة مقارنة بطرق عمل الريليهات الداخلية العادية.

مقارنة بين النوعين

م	الفرق	الريليهات الداخلية العادية	الريليهات الخاصة
١	كيفية العمل	تعمل حسب شروط وضعت فى البرنامج	تعمل حسب شروط وضعت فى الوحدة وغير قابلة للتعديل
٢	كيفية الإيقاف	تقف حسب شروط وضعت فى البرنامج	تقف حسب شروط وضعت فى الوحدة وغير قابلة للتعديل

تعمل الريليهات الخاصة أى الريليهات المبرمجة بطريقة محددة مسبقاً من قبل وحدة البرمجة, فمثلاً توجد:

- ريليهات تعمل مرة كل نصف ثانية.
- ريليهات تعمل لمرة واحدة فقط.
- ريليهات تعمل مرة كل دورة للبرنامج.
- ريليهات تعمل فى حالة أعطال.
- ريليهات تعمل مرة كل نصف ساعة.
- ريليهات تعمل فى حالة فصل البطارية.
- ريليهات تعمل فى حالة أجبار أى عنوان على العمل أو الإيقاف.

شرح لبعض مفاتيح الريليهات الخاصة:

م	أسم الريليه	شرح طريقة عمل الريليه الخاص
١	SM0.0	يغلق هذا المفتاح دائماً.
٢	SM0.1	يغلق هذا المفتاح فقط فى أول Cycle.
٣	SM0.2	يغلق هذا المفتاح فى حالة فقدان أى قيمة من قيم الـ Retentive.
٤	SM0.3	يغلق هذا المفتاح لزمن Cycle عندما تتحول وحدة الـ PLC إلى العمل.
٥	SM0.4	يغلق هذا المفتاح لثلاثين ثانية ويفتح لثلاثين ثانية أخرى.
٦	SM0.5	يغلق هذا المفتاح لنصف ثانية ويفتح لنصف ثانية أخرى.
٧	SM0.6	يغلق هذا المفتاح لزمن cycle ويفتح لزمن cycle أخر.
٨	SM0.7	يغلق هذا المفتاح عندما يكون المفتاح فى وضع run و يفتح عندما يكون فى وضع term.
٩	SM1.0	يغلق هذا المفتاح عندما يكون الناتج الخاص بأى عملية حسابية يساوى صفر.
١٠	SM1.1	يغلق هذا المفتاح عندما يكون الناتج أكبر من حجم الذاكرة المستخدمة.
١١	SM1.2	يغلق هذا المفتاح عندما يكون الناتج الخاص بأى عملية حسابية سالب.
١٢	SM1.3	يغلق هذا المفتاح عندما يتم القسمة على صفر فى أى عملية حسابية.
١٣	SM4.7	يغلق هذا المفتاح عندما يتم تطبيق أمر Force على أى عنوان.
١٤	SM5.0	يغلق هذا المفتاح عندما تحدث أى أعطال بخصوص وحدة المدخلات أو المخرجات.
١٥	SM1.6	يغلق هذا المفتاح عندما يتم تحويل رقم BCD غير صحيح إلى Binary.

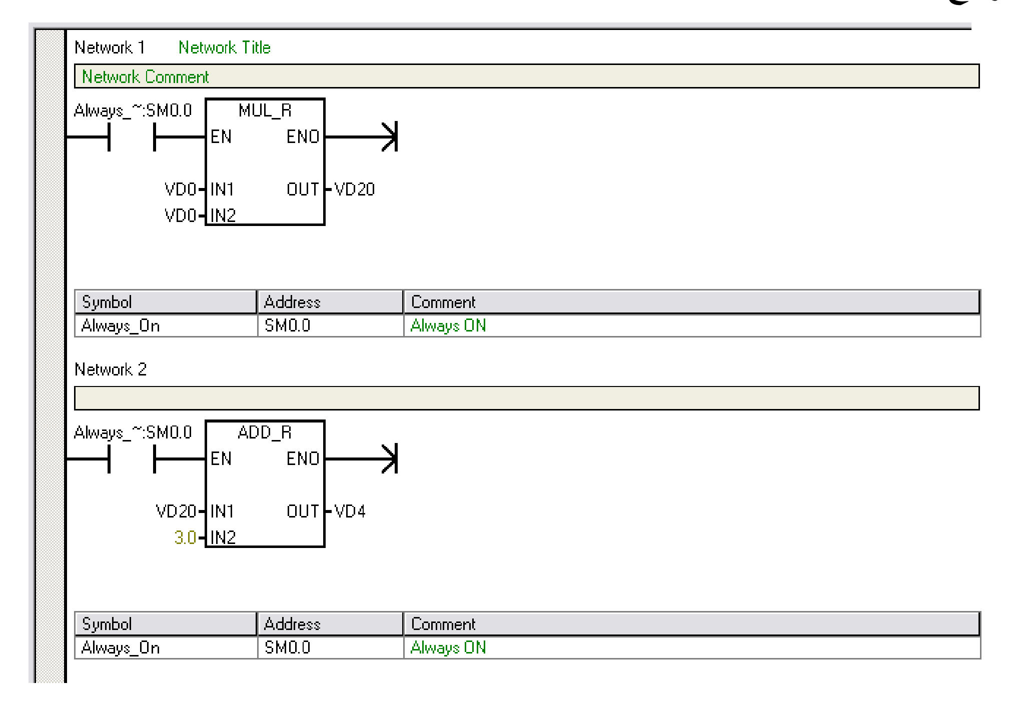
١٦	SM1.7	يغلق هذا المفتاح عندما لا يمكن تحويل رقم ASCII إلى ما يماثله في الـ Hexadecimal.
١٧	SMB6	تحتوى على رقم الـ CPU.
١٨	SMB28	تحتوى على قيمة الـ Analog Adjustment الأول.
١٩	SMB29	تحتوى على قيمة الـ Analog Adjustment الثانى.
٢٠	SMW22	يحتوى على قيمة الـ Scan time الخاصة بأخر Cycle.
٢١	SMW24	يحتوى على أقل قيمة للـ Scan time تم الوصول لها أثناء البرنامج.
٢٢	SMW26	يحتوى على أكبر قيمة للـ Scan time تم الوصول لها أثناء البرنامج.

تمارين عملية:

قم تنفيذ برنامج لتمثيل المعادلة التالية: ص = س^{٢} + ٣

عدد الدخل	**نوع الدخل**	**أسم الدخل**
لا يوجد	لا يوجد	لا يوجد
عدد العمليات الحسابية	**نوع العمليات الحسابية**	**أسم العمليات الحسابية**
١	ADD_R	ADD_R
٢	MUL_R	MUL_R
عدد المتغيرات	**نوع المتغيرات**	**أسم المتغيرات**
١	Dword, Real	VD0 (س)
٢	Dword, Real	VD4 (ص)
٣	Dword, Real	VD20
عدد الخرج	**نوع الخرج**	**أسم الخرج**
لا يوجد	لا يوجد	لا يوجد

البرنامج:



ملاحظة:

فبدلاً من وضع مفاتيح يمكن استخدام مفتاح من ضمن مفاتيح الريليهات الخاصة الذى يغلق دائماً.

تمارين عملية:

قم بتنفيذ برنامج لتمثيل المعادلة التالية بحيث أن تعمل الأصوات التحذيرية بطريقة متقطعة فى حالة:

- القسمة على صفر.
- ناتج سالب.
- ناتج صفر.
- ناتج أكبر من الذاكرة.

المعادلة:

ص = ٢س^٢ + ٤س + ٣

المدخلات:

عدد الدخل	**نوع الدخل**	**أسم الدخل**
لا يوجد	لا يوجد	لا يوجد

العمليات الحسابية:

عدد العمليات الحسابية	**نوع العمليات الحسابية**	**أسم العمليات الحسابية**
١	ADD_R	ADD_R
٢	MUL_R	MUL_R

الريليهات الخاصة:

عدد الريليهات الخاصة	**نوع الريليهات الخاصة**	**أسم الريليهات الخاصة**
١	Bit	SM0.0
٢	Bit	SM0.5
٣	Bit	SM1.0
٤	Bit	SM1.1
٥	Bit	SM1.2
٦	Bit	SM1.3

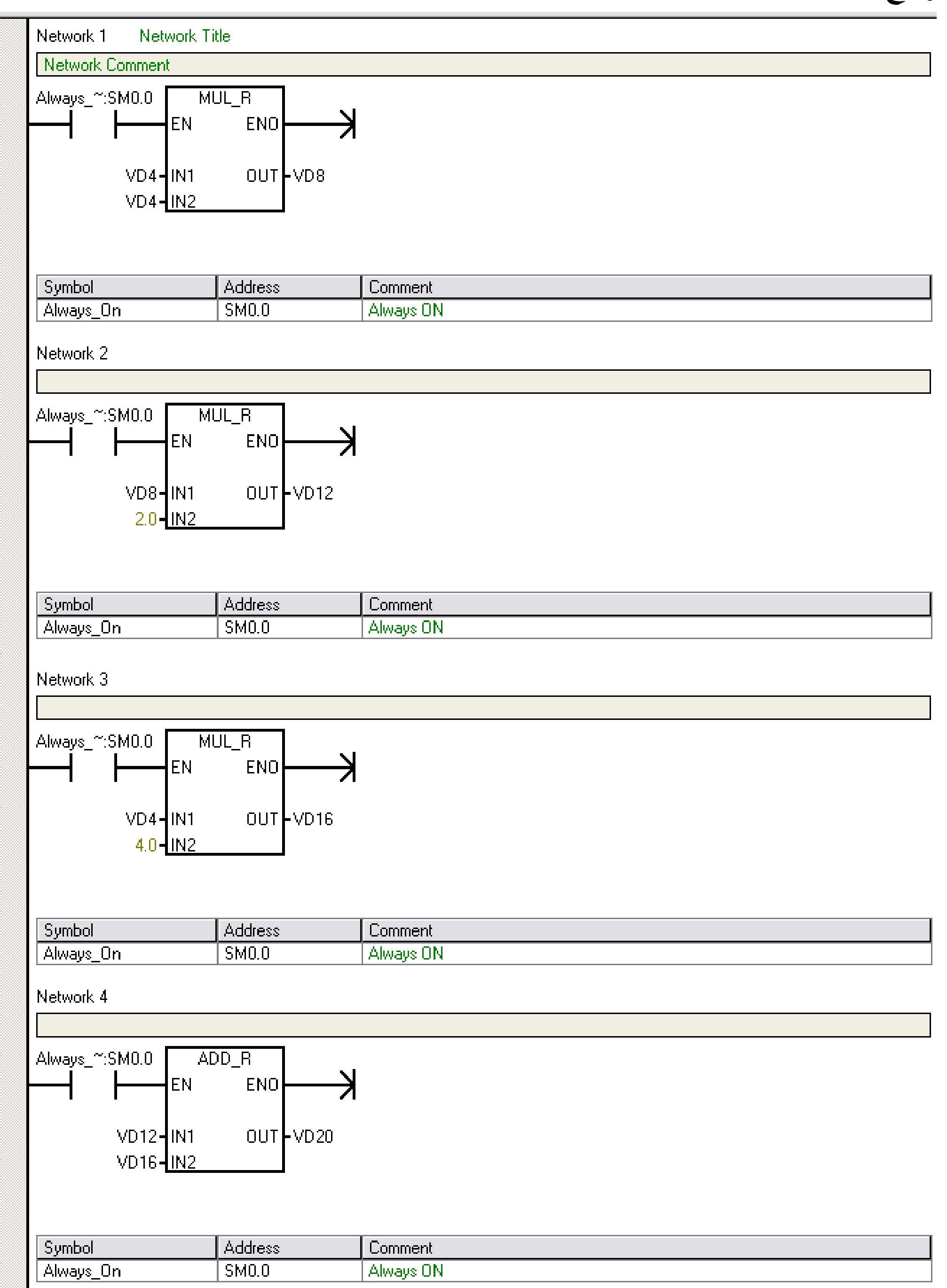
المتغيرات:

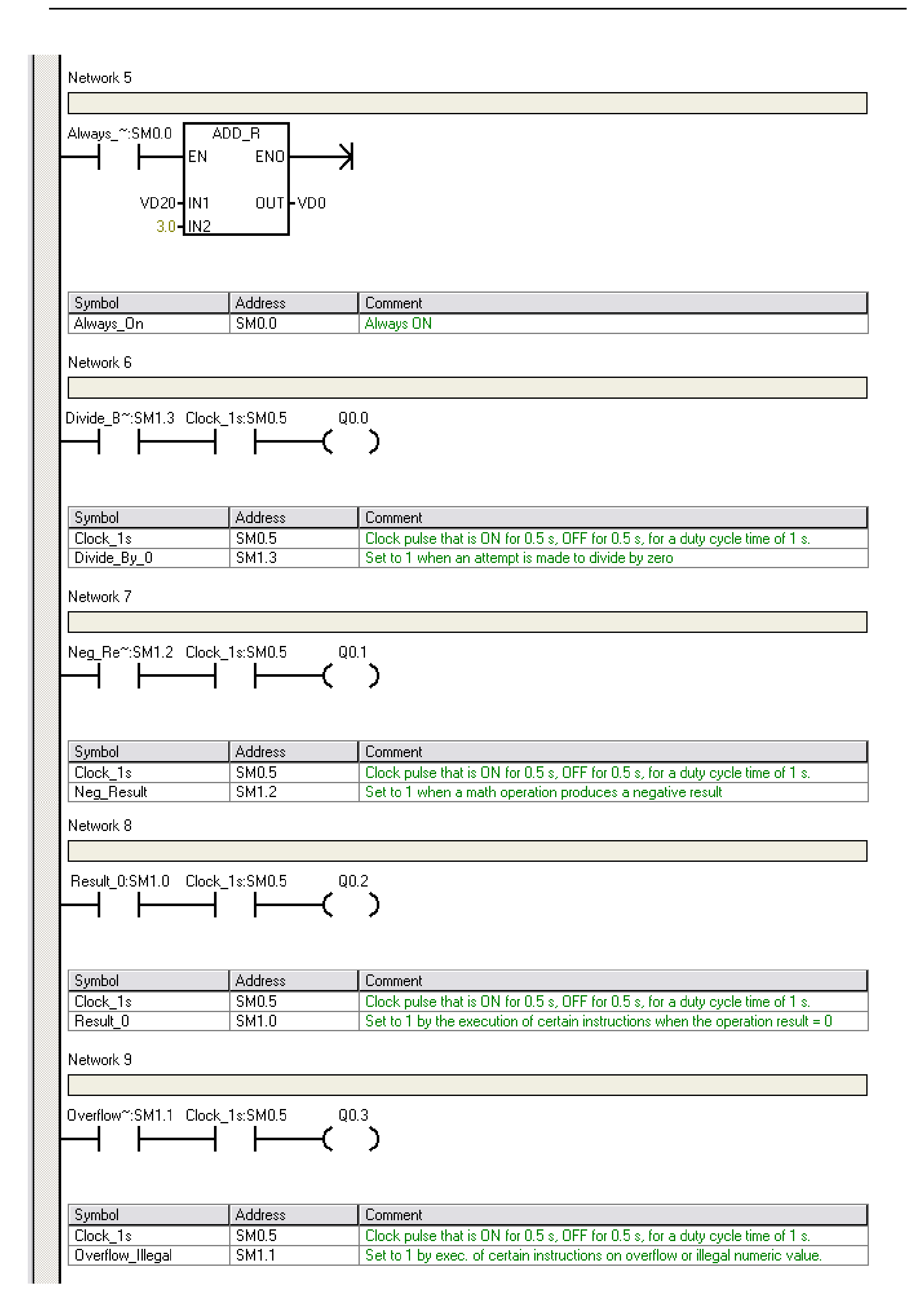
عدد المتغيرات	نوع المتغيرات	أسم المتغيرات
١	Dword, Real	VD0 (ص)
٢	Dword, Real	VD4 (س)
٣	Dword, Real	VD8 (س٢)
٤	Dword, Real	VD12 (٢س٢)
٥	Dword, Real	VD16 (٤س)
٦	Dword, Real	VD20 (٢س٢ + ٤س)

المخرجات:

عدد الخرج	نوع الخرج	أسم الخرج
١	إنذار (القسمة على صفر).	Q0.0/K1M
٢	إنذار (ناتج سالب).	Q0.1/K2M
٣	إنذار (ناتج صفر).	Q0.2/K3M
٤	إنذار (ناتج أكبر من الذاكرة).	Q0.3/K4M

البرنامج:


Network 1 Network Title
Network Comment
Always_~:SM0.0
MUL_R
EN ENO
VD4 IN1 OUT VD8
VD4 IN2
Symbol | Address | Comment
Always_On | SM0.0 | Always ON
Network 2
Always_~:SM0.0
MUL_R
EN ENO
VD8 IN1 OUT VD12
2.0 IN2
Symbol | Address | Comment
Always_On | SM0.0 | Always ON
Network 3
Always_~:SM0.0
MUL_R
EN ENO
VD4 IN1 OUT VD16
4.0 IN2
Symbol | Address | Comment
Always_On | SM0.0 | Always ON
Network 4
Always_~:SM0.0
ADD_R
EN ENO
VD12 IN1 OUT VD20
VD16 IN2
Symbol | Address | Comment
Always_On | SM0.0 | Always ON



Symbol	Address	Comment
Always_On	SM0.0	Always ON

Symbol	Address	Comment
Clock_1s	SM0.5	Clock pulse that is ON for 0.5 s, OFF for 0.5 s, for a duty cycle time of 1 s.
Divide_By_0	SM1.3	Set to 1 when an attempt is made to divide by zero

Symbol	Address	Comment
Clock_1s	SM0.5	Clock pulse that is ON for 0.5 s, OFF for 0.5 s, for a duty cycle time of 1 s.
Neg_Result	SM1.2	Set to 1 when a math operation produces a negative result

Symbol	Address	Comment
Clock_1s	SM0.5	Clock pulse that is ON for 0.5 s, OFF for 0.5 s, for a duty cycle time of 1 s.
Result_0	SM1.0	Set to 1 by the execution of certain instructions when the operation result = 0

Symbol	Address	Comment
Clock_1s	SM0.5	Clock pulse that is ON for 0.5 s, OFF for 0.5 s, for a duty cycle time of 1 s.
Overflow_Illegal	SM1.1	Set to 1 by exec. of certain instructions on overflow or illegal numeric value.

الشرح:

- **الجزء الخاص بالمعادلة الرياضية:**

:Network1

حيث يقوم بضرب قيمة VD4 فى VD4 ثم يضع الناتج فى VD8.

:Network2

حيث يقوم بضرب قيمة VD8 فى 2.0 ثم يضع الناتج فى VD12.

:Network3

حيث يقوم بضرب قيمة VD4 فى 4.0 ثم يضع الناتج فى VD16.

:Network4

حيث يقوم بجمع قيمة VD12 مع VD16 ثم يضع الناتج فى VD20.

:Network5

حيث يقوم بجمع قيمة VD12 مع 3.0 ثم يضع الناتج فى VD0.

- **الجزء الخاص بالريليهات الخاصة:**

:Network6

فى حاله القسمة على صفر سوف يعمل الإنذار Q0.0 بطريقة متقطعة.

:Network7

فى حاله أن الناتج قيمة سالبة سوف يعمل الإنذار Q0.1 بطريقة متقطعة.

:Network8

فى حاله أن الناتج يساوى صفر سوف يعمل الإنذار Q0.2 بطريقة متقطعة.

:Network9

فى حاله خطاء فى القيمة الرقمية للذاكرة سوف يعمل الإنذار Q0.3 بطريقة متقطعة.

الباب العاشر

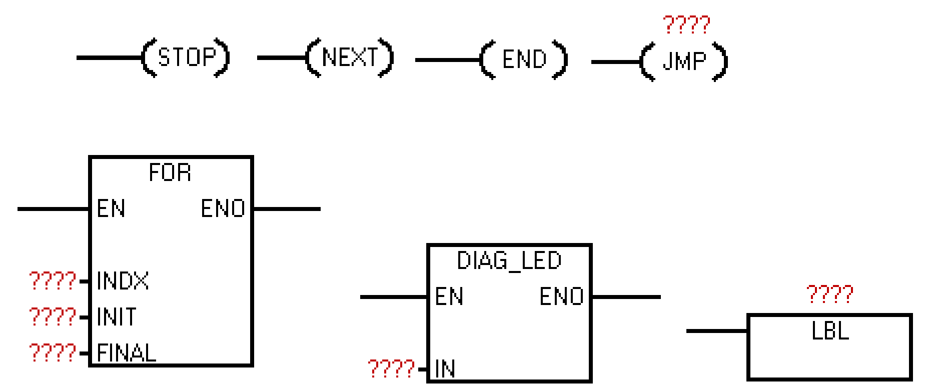
برامج التحكم

- شرح برامـــــــــج التحكم.
- محتويات برامـــــــج التحكم.
- شرح عمليــــــــــــة END.
- شرح عمليــــــــــة STOP.
- شرح عمليــــــــــــة RET.
- شرح عمليـــة DIAG_LED.
- شرح عمليـــــة JMP/LBL.
- شرح عمليـــة FOR/NEXT.
- الأخـــــطاء الممكن التعرض لها.
- تمـــــــــارين عملية للتوضيح.

برامج التحكم:

تستخدم بعض الأوامر التالية للمساعدة فى البرمجة وليس بالشرط كجزء أساسى فى البرنامج.

END – STOP – JMP/LBL – FOR/NEXT – DIAGLED – RET



م	الأسم	الشرح	الشكل
١	END	يستخدم أمر END للتحكم بتنفيذ جزء معين حيث أنه يتحكم فى حجم الـ Cycle.	END
٢	STOP	يستخدم أمر STOP لإيقاف وحدة الـ PLC.	STOP
٣	DIAG_LED	يستخدم أمر DIAG_LED للتحكم بلمبة الـ S.F.	DIAG_LED EN ENO ???? IN

٤	JMP/LBL	يستخدم أمر JMP/LBL للتحكم بتنفيذ أو إلغاء جزء معين فى وسط ال Cycle.	???? JMP ???? LBL
٥	FOR/NEXT	يستخدم أمر FOR/NEXT للتحكل بقراءة جزء معين فى البرنامج أكثر من مرة قبل الأنتهاء من ال Cycle.	NEXT FOR EN ENO ????-INDX ????-INIT ????-FINAL
٦	RET	يستخدم أمر RET للخروج من صفحة البرمجة الفرعية قبل الأنتهاء من تنفيذ البرنامج الفرعى بالكامل.	RET

شرح كل نوع ورسم تمرين عملى للتوضيح

١- END:

كما سبق وذكرنا أن أمر END يستخدم للتحكم بتنفيذ جزء معين فى ال Cycle, فمثلاً أذا كان البرنامج متكون من أربع أفرع فهذا يعنى أن أثناء ال Cycle يتم قراءة الأربع أفرع بالكامل ولكن إذا تم وضع أمر END فى الفرع الثالث فهذا يعنى أن أثناء ال Cycle يتم قراءة الثلاث أفرع الأولى فقط أما قراءة ما يوجد فى الفرع الرابع سوف يبقى مرهون بأمر END فإذا كان لا يعمل فسيكون البرنامج طبيعى جداً كما أن أمر END لم يرسم من الأساس إما إذا كان يعمل فسيبقى حاله البرنامج المرسوم فى الفرع الرابع على ما هو علية إلى أن يلغى أمر END مرة أخرى وتتم قراءة الفرع الرابع مرة أخرى.

تمرين عملى باستخدام أمر END:

مكينة تعمل من مكان واحد وتوجد لمبة توضيء فوق المكينة بشرط أن تعمل هذه اللمبة فقط فى الليل.

عدد الدخل	نوع الدخل	أسم الدخل
١	n.c.	I0.0/S1
٢	n.o.	I0.1/S2
٣	n.o.	I0.4/S3 (خلية ضوئية)
عدد التحكمات البرمجية	**نوع التحكمات البرمجية**	**أسم التحكمات البرمجية**
١	END	END
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتكتور	Q0.1/K1M
٢	اللمبة	Q0.5/K2M

البرنامج:

Network 1 Network Title
Network Comment
I0.1 I0.0 Q0.1
Q0.1

Network 2
I0.4 END

Network 3
Q0.1 Q0.5

ملاحظة:

فى حقيقة الأمر لا يمكن أن نقول بأن أمر END يتحكم فى حجم الدورة **cycle** لأن هذا سيخالف ما قد تم شرحة فى الجزء الأول وهو ما يأكد بأن الـ **cycle time** ثابت لا يتغير, لذالك الذى حدث فى حاله استخدام أمر END لم يكن سوى أن البرنامج قد تظاهر بأنة لم يرى الفرع الرابع لكنة فى جوهر الأمر قد مر على جميع الأفرع بالكامل.

٢- STOP:

كما سبق وذكرنا أن أمر STOP يستخدم لإيقاف وحدة الـ PLC, فأى أن كان عدد الأفرع المتكون منها البرنامج فأنه يتم وضع أمر STOP فى أى فرع من فروع البرنامج فأنه يوقف وحدة الـ PLC, حيث يتم استخدام هذا الأمر فى حالات الطوارئ القسوة التى تستدعى توقف الـ PLC بالكامل, لا يهم كثيراً فى أى فرع سيتم وضع أمر STOP لأن فى جميع الأحوال سوف يتوقف الـ PLC.

تمرين عملى باستخدام أمر STOP:

مكينة تعمل من مكان واحد وفى حالة وجود حريق بالمصنع يجب لوحدة الـ PLC التوقف عن العمل.

عدد الدخل	نوع الدخل	أسم الدخل
١	n.c.	I0.0/S1
٢	n.o.	I0.1/S2
٣	n.o.	I0.7/S3 (حساس حريق)
عدد التحكمات البرمجية	**نوع التحكمات البرمجية**	**أسم التحكمات البرمجية**
١	STOP	STOP
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتكتور	Q0.1/K1M

البرنامج:

Network 1 Network Title

Network Comment

I0.1 I0.0 Q0.1

Q0.1

Network 2

I0.7 (STOP)

ملاحظة:

فى حالة استخدام أمر STOP يجب التأكد من إرسال أشارة غير مستمرة لتنفيذ هذا الأمر وذالك لأن فى حاله إرسال إشارة مستمرة على هذا الأمر فهذا يعنى أن كل مرة سوف تقوم فيه بتشغيل وحدة البرمجة سوف تتوقف تلقائياً فلذالك يفضل وضع مفتاح positive edge قبل أمر STOP مباشراً لضمان عدم استمرار الإشارة لفترة طويلة.

٣- DIAG_LED:

كما سبق وذكرنا أن أمر DIAG_LED يستخدم للتحكم بلمبة الـ S.F.,مع ملاحظة أن اللمبة قد تستخدم من قبل وحدة الـ PLC فى حالة أعطال فاضحة كما سوف نشرح بعد قليل وقد تستخدم اللمبة أيضاً من قبل المبرمج كما سوف نشرح فى المثال الحالى.

تضاء اللمبة فقط أذا كانت قيمة الـ IN مختلفة عن صفر, فمثلاً أذا كانت القيمة هى ٧ فستعمل اللمبة.

تمرين عملى باستخدام أمر DIAG_LED:

مكينة تعمل من مكان واحد وفى حالة وجود أى أشارة من الـ Over load فأنة يفصل المكينة ويضيء لمبة الـ S.F. باللون الأصفر.

عدد الدخل	نوع الدخل	أسم الدخل
١	n.c.	I0.0/S1
٢	n.o.	I0.1/S2
٣	n.o.	I0.2/S3(Over Load)
عدد التحكمات البرمجية	**نوع التحكمات البرمجية**	**أسم التحكمات البرمجية**
١	DIAG_LED	DIAG_LED
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتكتور	Q0.1/K1M

البرنامج:

Network 1 Network Title
Network Comment
I0.1 I0.0 I0.3 Q0.1
Q0.1

Network 2
I0.3 DIAG_LED
EN ENO
7 IN

ملاحظة:

- تضاء اللمبة بالون الأصفر فى حاله استخدامها من قبل المبرمج كما فى التمرين الحالى.
- تضاء اللمبة بالون الأصفر فى حاله تطبيق أمر FORCE على أى عنوان كما سبق وشرحنا فى الفصل الثانى من هذا الكتاب.
- تضاء اللمبة بالون الأحمر فى حاله استخدامها من قبل وحدة الـ PLC فى حالة الأعطال الفاضحة كما سيتم التوضيح فالكتاب التالى الخاص بالأعطال والتمرين العملية.

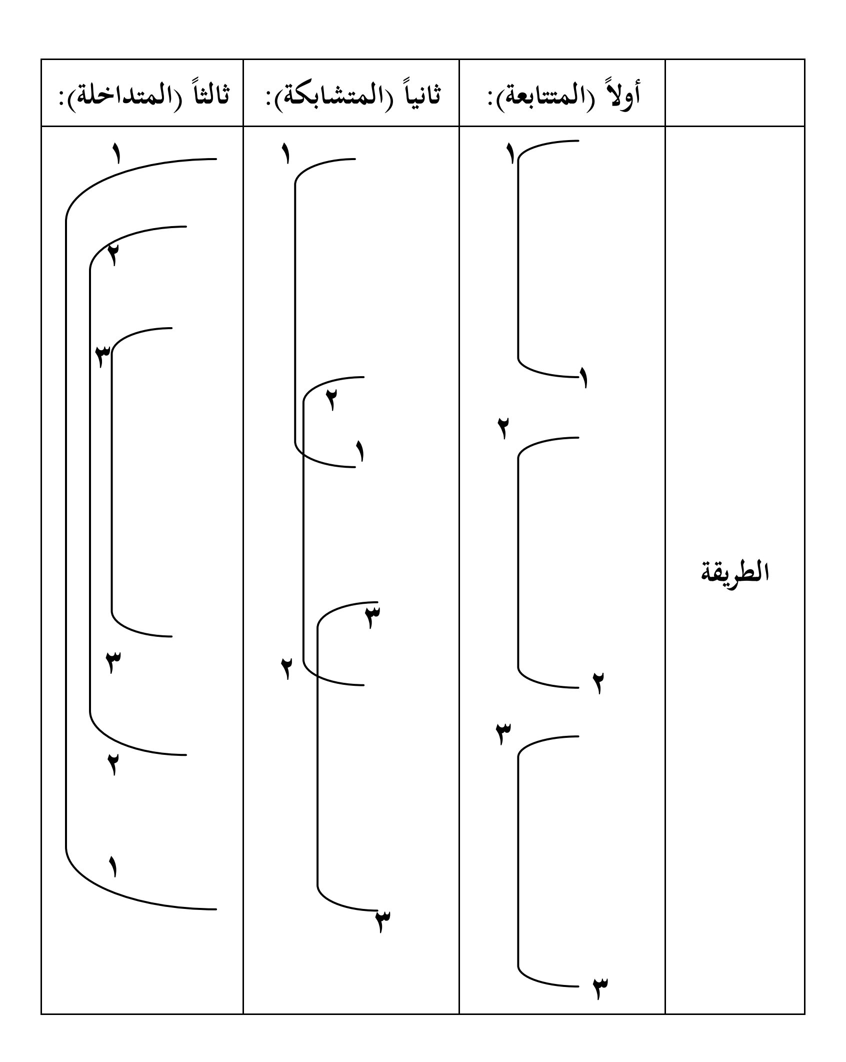
٤- JMP/LBL:

كما سبق وذكرنا أن أمر JMP/LBL يستخدم للتحكم بتنفيذ جزء معين فى وسط الـ Cycle, بحيث ان هذا الأمر يقوم بتنفيذ كل البرنامج ماعدا الأفرع الموجودة بين الـ JMP و الـ LBL مع ملاحظة أن يتم كتابة نفس الرقم على الـ JMP و الـ LBL.

ملاحظة:

- لمعرفة أى JMP يخص أى LBL يكتب نفس الرقم على الأثنان.
- الأرقام المستخدمة تبدأ من صفر إلى ٢٥٥.
- يمكن أن يتكرر الـ JMP و الـ LBL أكثر من مرة ولكن عدد الـ JMP يجب أن يساوى عدد الـ LBL.
- شرط أن يكون الـ JMP فوق الـ LBL وليس العكس.
- يمكن لـ JMP و الـ LBL أن يتكرر بطرق مختلفة: (متتابعة – متشابكة – متداخلة).

الأنواع الثلاثة: المتتابعة – المتشابكة – المتداخلة



	أولاً (المتتابعة):	ثانياً (المتشابكة):	ثالثاً (المتداخلة):
الطريقة			

تمرين عملى باستخدام أمر JMP-LBL:

محرك يعمل فى اتجاهين, ويوجد أيضاً فلاشر يعمل اختيارياً مع المحرك أى أنه يمكن إلغاء الفلاشر.

عدد الدخل	نوع الدخل	أسم الدخل
١	n.c.	I0.0/S1
٢	n.o.	I0.1/S2
٣	n.o.	I1.1/S3
٤	n.c.	I2.2/S4
عدد التحكمات البرمجية	**نوع التحكمات البرمجية**	**أسم التحكمات البرمجية**
١	JMP/LBL	4
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتكتور	Q0.1/K1M
٢	كونتكتور	Q0.2/K2M
٣	لمبة	Q1.1/K3M
٤	لمبة	Q1.2/K4M

ملاحظة:

- فى حالة استخدام أمر JMP-LBL يتطلب دائماً أن يتم رسم الجزء الخاص بال JMP فوق الجزء الخاص بال LBL وذالك لأنه حين يعمل ال JMP فأنه سيقفز إلى أسفل حتى يصل إلى ال LBL المعنون بنفس عنوان ال JMP.
- فى حالة استخدام أمر JMP-LBL ولكن من دون الألتزام بالشروط السابق ذكرها أى أن فى حالة أن يتم رسم الجزء الخاص بال JMP أسفل الجزء الخاص بال LBL ستحدث مشكلة وذالك لأنه حين يعمل ال JMP فأنه سيقفز إلى الأعلى حتى يصل إلى ال LBL المعنون بنفس رقم ال JMP من ما يستدعى إلى الدوران عكس الحركة الطبيعية للبرنامج فتحدث المشكلة وسوف يتم شرح الأعطال بالتفصيل فى الكتاب التالى الخاص بالأعطال والتمارين العملية.

البرنامج:

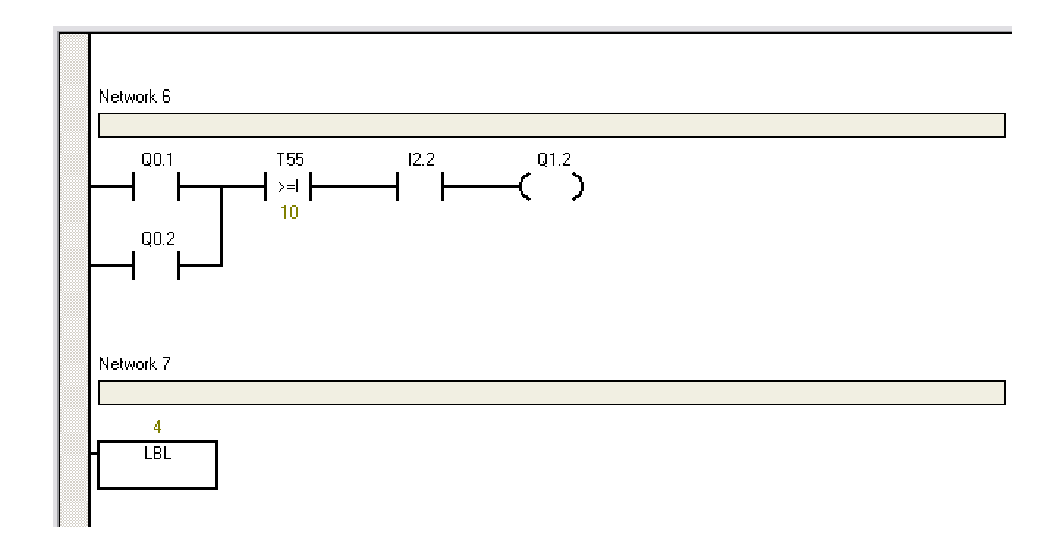
```
Network 1    Network Title
Network Comment
  I0.1        I0.0        Q0.2        Q0.1
--| |----+----| |---------|/|---------(   )
  Q0.1   |
--| |----+

Network 2    Network Title
Network Comment
  I0.2        I0.0        Q0.1        Q0.2
--| |----+----| |---------|/|---------(   )
  Q0.2   |
--| |----+

Network 3
  I1.1        4
--| |--------(JMP)

Network 4
  Q0.1        T55         I2.2                    T55
--| |----+----|<>I|-------| |----------------[IN      TON]
  Q0.2   |     20                           1-[PT   100 ms]
--| |----+

Network 5
  Q0.1        T55         I2.2        Q1.1
--| |----+----|<=I|-------| |---------(   )
  Q0.2   |     10
--| |----+
```



٥- RET:

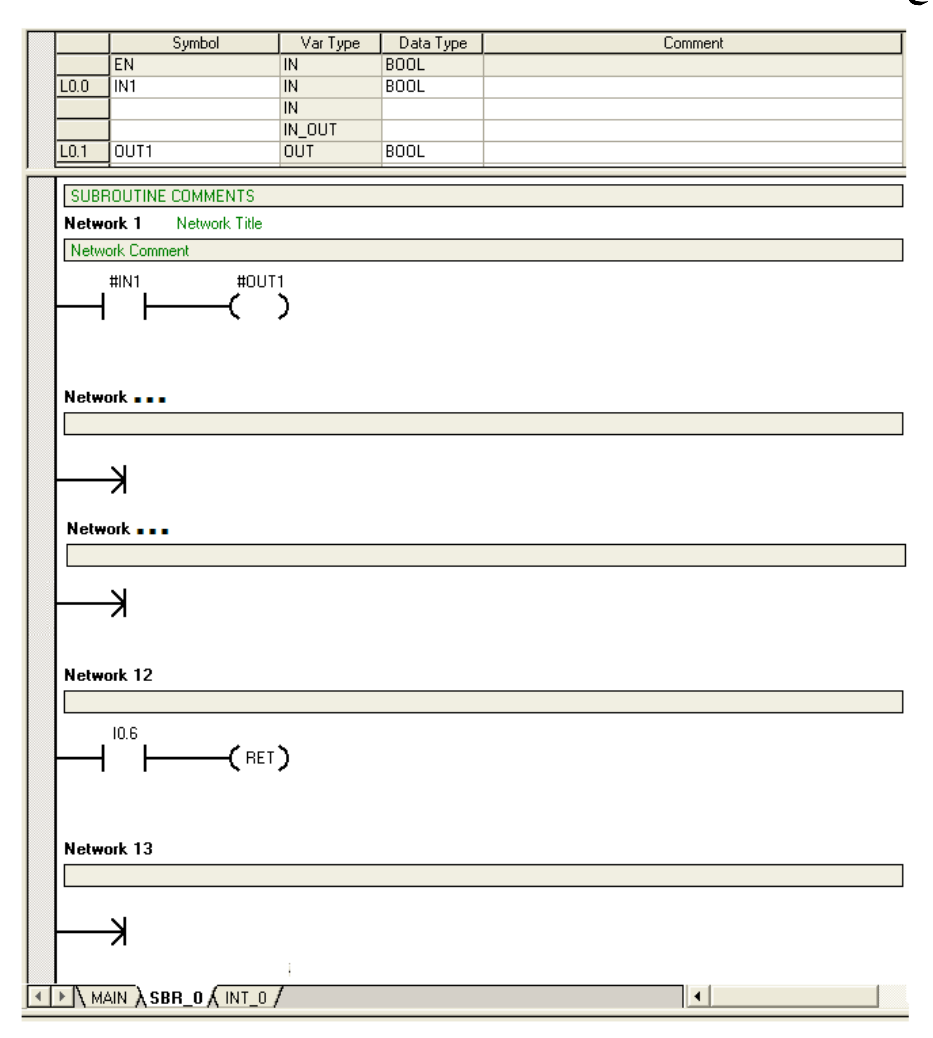
كما سبق وذكرنا أن أمر RET يستخدم للرجوع من صفحة البرمجة الفرعية إلى صفحة البرمجة الرئيسية قبل الأنتهاء من تنفيذ البرنامج الفرعى بالكامل أو حتى بعد الأنتهاء من تنفيذ البرنامج بالكامل.

تمرين عملى باستخدام أمر RET:

تمرين يحتوى على برنامج فرعى وفى حاله الضغط على المفتاح I0.1 فيخرج من صفحة البرنامج الفرعى إلى البرنامج الرئيسى فى الحال دون أن يكمل حتى يصل إلى الفرع الأخير.

عدد الدخل	**نوع الدخل**	**أسم الدخل**
١	n.c.	I0.1/S1
عدد التحكمات البرمجية	**نوع التحكمات البرمجية**	**أسم التحكمات البرمجية**
١	RET	RET

البرنامج:



	Symbol	Var Type	Data Type	Comment
	EN	IN	BOOL	
L0.0	IN1	IN	BOOL	
		IN		
		IN_OUT		
L0.1	OUT1	OUT	BOOL	

توضيح:

ليس الفكرة فى البرنامج بل الفكرة هى توضيح انه يمكن أن يتم الخروج من البرنامج الفرعى دون الانتظار حتى نهاية الأفرع بالكامل وذالك يحدث فى حالة تشغيل أمر RET فقط.

٦- FOR/NEXT:

كما سبق وذكرنا أن أمر FOR/NEXT يستخدم للتحكم بقراءة جزء معين فى البرنامج أكثر من مرة قبل الأنتهاء من الـ Cycle .

تمرين عملى باستخدام أمر FOR/NEXT:

تمرين يحتوى على برنامج معين وفى حاله الضغط على المفتاح I0.1 فأنه يتم قراءة الفرع الثانى والثالث عشرة مرات متتالية قبل الأنتهاء من الدورة Cycle.

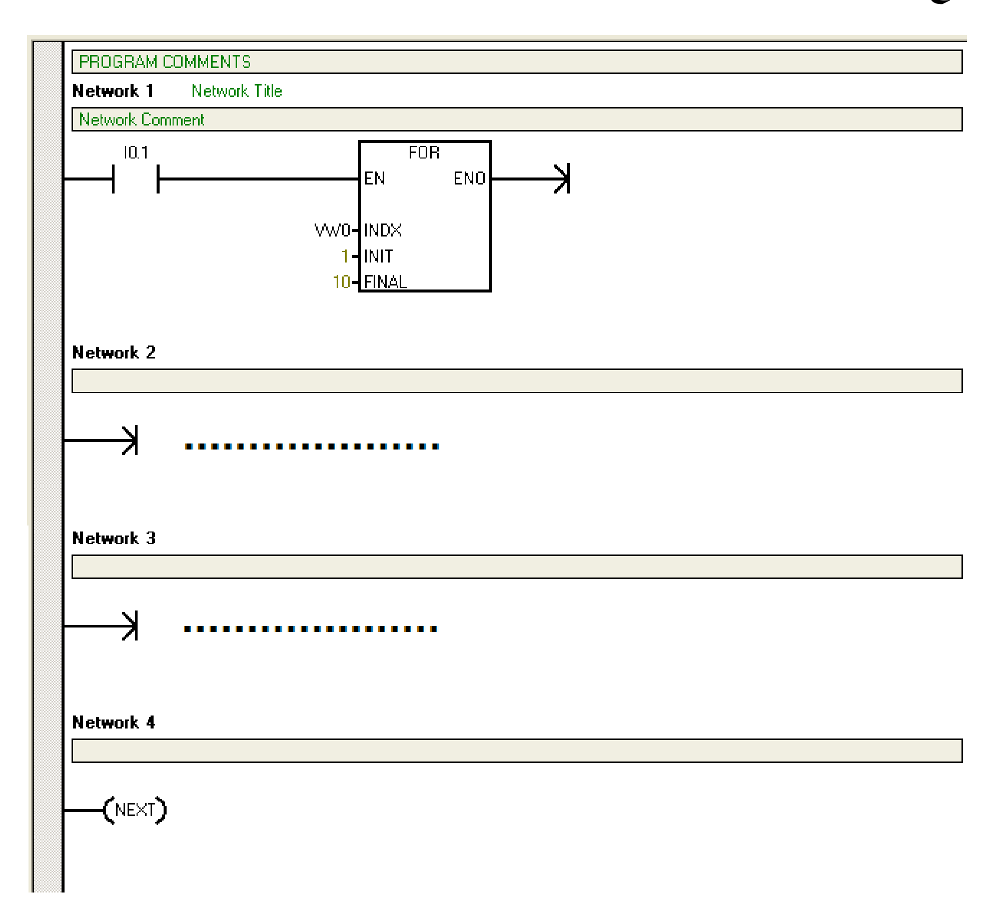
عدد الدخل	**نوع الدخل**	**أسم الدخل**
١	n.c.	I0.1/S1
عدد التحكمات البرمجية	**نوع التحكمات البرمجية**	**أسم التحكمات البرمجية**
١	FOR/NEXT	FOR/NEXT

فمثلاً فى التمرين التالى تم تسجيل قيمة ١ فى INIT بينما تم تسجيل قيمة ١٠ فى FINAL فا فى هذه الحالة سيتم المرور على الأفرع المتواجدة بين FOR و NEXT عشرة مرات بحيث أن قيمة INDX سوف يبدأ من واحد إلى عشرة

ملاحظة:

نظراً لأن قيمة INDX سوف تتغير عدة مرات حيث سيضاف ١ إلى القيمة كل دورة فلهذا لا يمكن تحديد أى قيمة مسبقة فى هذا المكان بل سيتم كتابة متغير بحجم word حتى يتيح إلى المحتوى أن يتغير بسهوله.
عند المرور على أمر FOR للمرة الأولى تكون قيمة الـ VW0 ← [١] ثم فى المرة الثانى يضاف إلى الرقم واحد فتصبح قيمة الـ VW0 ← [٢] وهكذا[٣], [٤], [٥], [٦], [٧], [٨], [٩], [١٠]

البرنامج:



أى أن كان البرنامج المرسوم فى الفرع الثانى والثالث فأنه فى حاله غلق المفتاح I0.1 سوف يتم قراءة الأفرعين الثانى والثالث عشرة مرات قبل الانتهاء من الدورة الواحدة.

الباب الحادى العاشر

المحولات

- شرح المحـــــــــــــــــــــــــــــولات.
- أنواع المحـــــــــــــــــــــــــــــولات.
- محـــــــــــــــــــــــــولات لتغير الحجم.
- محـــــــــــــــــــــولات لتغير الـ format.
- الفـــــــــــــــــرق بين word و integer.
- الفـــــــــــــرق بين Dword و Dinteger.
- المفاتيح المستخدمة مع المحــــــــــــــولات.
- الأخطاء الممكن التعرض لها باستخدام المحـولات.
- تمـــــــــــــــــــــــــارين عملية للتوضيح.

المحولات:

تستخدم المحولات للتحويل بثلاث طرق مختلفة:

- **من حجم إلى حجم مختلف.**

فهذا يتحقق في حالة التحويل من Byte إلى Word مثلاً, حيث يتم تغير الحجم من 8bits إلى 16bits.

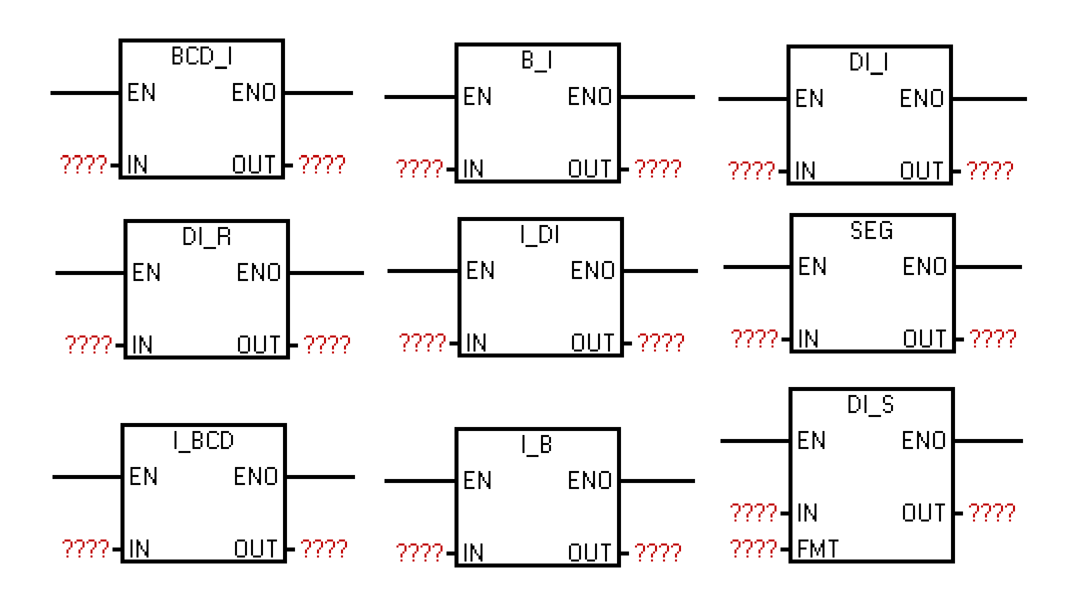
- **من format إلى format أخر.**

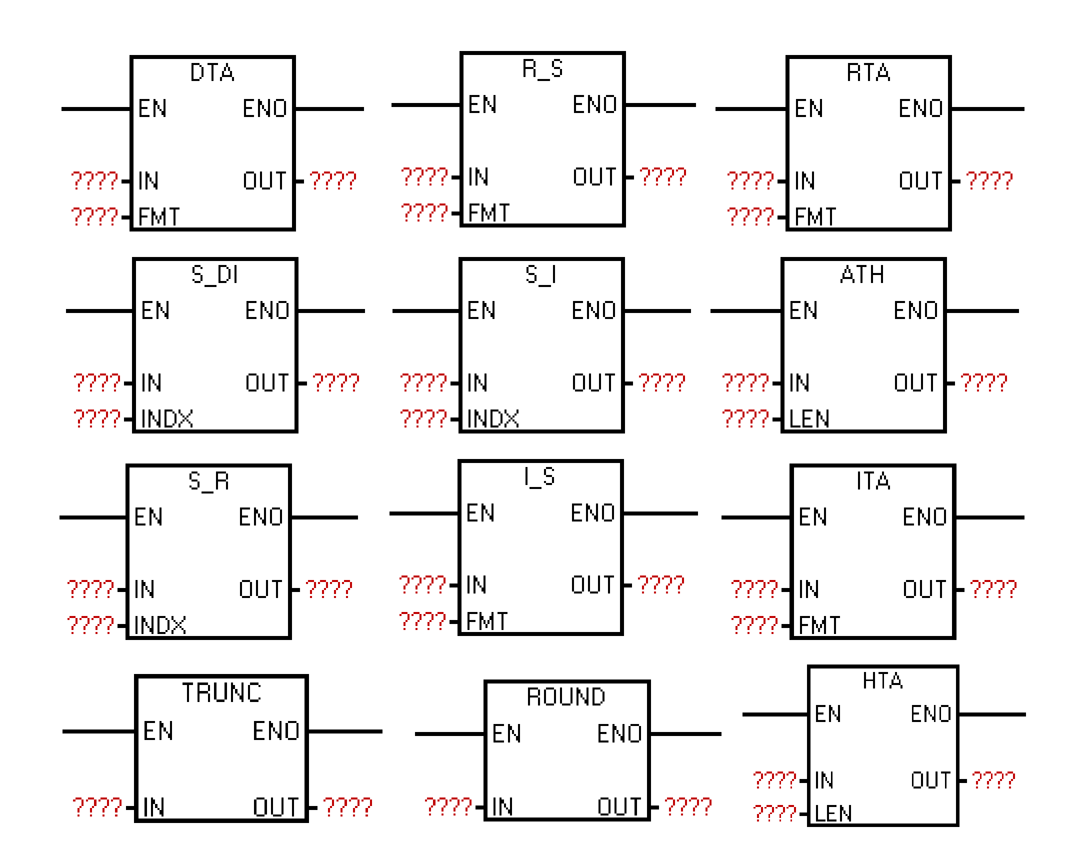
فهذا يتحقق في حالة التحويل من Dinteger إلى Real مثلاً, حيث يتم تغير الـ format من أرقام صحيحة إلى أرقام عشرية دون تغير حجم الذاكرة.

- **من حجم و format إلى حجم و format أخر.**

فهذا يتحقق في حالة التحويل من Byte إلى Integer مثلاً حيث يتم تغير الحجم من 8bits بدون إشارة إلى 16bits بإشارة.

التحويلات المستخدمة في البرمجة:





شرح التحويلات المستخدمة فى البرنامج.

م	الأسم	الشرح	الشكل
١	B_I	يستخدم الـ B_I للتحويل من حجم Byte إلى حجم Word مع الحفاظ على نفس القيمة.	B_I EN ENO ???? IN OUT ????
٢	I_B	يستخدم الـ I_B للتحويل من حجم Word إلى حجم Byte بشرط أن لا تكون القيمة أكبر من أن تكتب على Byte .	I_B EN ENO ???? IN OUT ????

٣	I_DI	يستخدم الـ I_DI للتحويل من حجم Word إلى حجم Dword مع الحفاظ على نفس القيمة.	I_DI EN ENO ????-IN OUT-????
٤	I_S	يستخدم الـ I_S للتحويل من حجم Word إلى حجم Byte بحيث يقوم بتحويل الأرقام إلى أحرف.	I_S EN ENO ????-IN OUT-???? ????-FMT
٥	DI_I	يستخدم الـ DI_I للتحويل من حجم Dword إلى حجم Word بشرط أن لا تكون القيمة أكبر من أن تكتب على Word.	DI_I EN ENO ????-IN OUT-????
٦	DI_R	يستخدم الـ DI_R للتحويل من حجم Dword إلى حجم Dword ولكن يتم تحويل القيمة من رقم صحيح إلى رقم عشرى .	DI_R EN ENO ????-IN OUT-????
٧	DI_S	يستخدم الـ DI_S للتحويل من حجم Dword إلى حجم Byte بحيث يقوم بتحويل الأرقام إلى أحرف.	DI_S EN ENO ????-IN OUT-???? ????-FMT
٨	BCD_I	يستخدم الـ BCD_I للتحويل من حجم Word إلى حجم Word ولكن مع تغير النظام من BCD إلى Integer.	BCD_I EN ENO ????-IN OUT-????

٩	I_BCD	يستخدم الـ I_ BCD للتحويل من حجم Word إلى حجم Word ولكن مع تغير النظام من Integer إلى BCD.	I_BCD EN ENO ????-IN OUT-????
١٠	ROUND	يستخدم الـ TRUNC للتحويل رقم عشرى بحجم Dword إلى رقم صحيح بحجم Dword وذالك بالتقريب إلى أقرب رقم صحيح.	ROUND EN ENO ????-IN OUT-????
١١	TRUNC	يستخدم الـ TRUNC للتحويل رقم عشرى بحجم Dword إلى رقم صحيح بحجم Dword ولذالك يتم مسح الرقم المكتوب بعد العلامة العشرية.	TRUNC EN ENO ????-IN OUT-????
١٢	R_S	يستخدم الـ R_S للتحويل من أرقام عشرية بحجم Dword إلى أحرف بحجم Byte.	R_S EN ENO ????-IN OUT-???? ????-FMT
١٣	ITA	يستخدم الـ ITA لتحويل من أرقام صحيحة بحجم Word إلى ما يعادلها في جدول ASCII بحجم Byte.	ITA EN ENO ????-IN OUT-???? ????-FMT
١٤	DTA	يستخدم الـ DTA لتحويل من أرقام صحيحة بحجم Dword إلى ما يعادلها في جدول ASCII بحجم Byte.	DTA EN ENO ????-IN OUT-???? ????-FMT

١٥	RTA	يستخدم الـ RTA للتحويل من أرقام عشرية بحجم Dword إلى ما يعادلها في جدول ASCII بحجم Byte.	RTA EN ENO ????-IN OUT-???? ????-FMT
١٦	ATH	يستخدم الـ ATH للتحويل من (أرقام, رموز أو أحرف في جدول ASCII) بحجم Byte إلى ما يعادلها في لغة Hexadecimal بحجم Byte أيضاً.	ATH EN ENO ????-IN OUT-???? ????-LEN
١٧	HTA	يستخدم الـ HTA للتحويل من أرقام بلغة Hexadecimal و بحجم Byte إلى ما يعادلها من (أرقام, رموز أو أحرف في جدول ASCII) بحجم Byte أيضاً.	HTA EN ENO ????-IN OUT-???? ????-LEN
١٨	S_I	يستخدم الـ S_I للتحويل من حجم Byte إلى حجم Word بحيث يقوم بتحويل الأحرف إلى أرقام صحيحية.	S_I EN ENO ????-IN OUT-???? ????-INDX
١٩	S_DI	يستخدم الـ S_DI للتحويل من حجم Byte إلى حجم Dword بحيث يقوم بتحويل الأحرف إلى أرقام صحيحية.	S_DI EN ENO ????-IN OUT-???? ????-INDX
٢٠	S_R	يستخدم الـ S_R للتحويل من حجم Byte إلى حجم Dword بحيث يقوم بتحويل الأحرف إلى أرقام عشرية.	S_R EN ENO ????-IN OUT-???? ????-INDX

٢١	SEG	يستخدم الـ SEG لإضاءة اللمبات الخاصة بالـ (SSD) SEVEN SEGMEND DISPLAY حسب الأرقام المكتوبة فى IN التى هى بحجم Byte.	SEG EN ENO ????-IN OUT-????

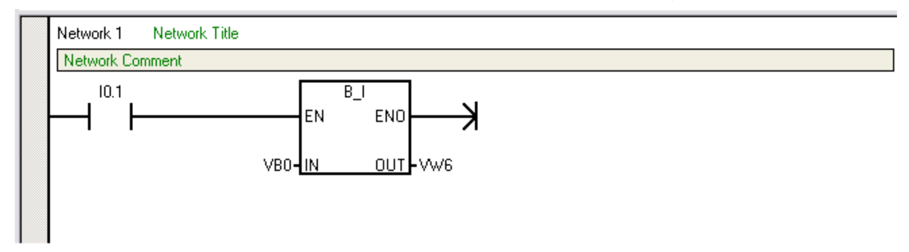
ملاحظة:

كيفية معرفة أذا كان الرقم الذى سيكتب على الذاكرة هو رقم صحيح أم رقم عشرى و كيفية معرفة النظام الذى سيستخدم لعرض القيمة.

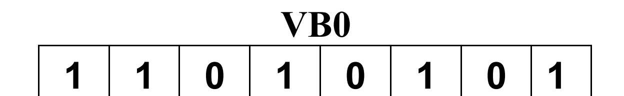
م	الأسم	المقصود	الشرح
١	W	WORD	هـذه الـذاكرة مكونـة مـن ١٦ bits ويكتـب عليهـا أرقـام صـحيحة دون إشارة.
٢	I	INTEGER	هـذه الـذاكرة مكونـة مـن ١٦ bits ويكتـب عليهـا أرقـام صـحيحة بإشارة موجبة أو سالبة.
٣	DW	DWORD	هـذه الـذاكرة مكونـة مـن ٣٢ bits ويكتـب عليهـا أرقـام صـحيحة دون إشارة.
٤	DI	DINTIGER	هـذه الـذاكرة مكونـة مـن ٣٢ bits ويكتـب عليهـا أرقـام صـحيحة بإشارة موجبة أو سالبة.
٥	R	REAL	هـذه الـذاكرة مكونـة مـن ٣٢ bits ويكتـب عليهـا أرقـام عشـرية بإشارة موجبة أو سالبة.
٦	S	STRING	هـذه الـذاكرة مكونـة مـن ٨ bits ويكتـب عليهـا أرقـام صـحيحة لتمثيل حرف واحد.

مثال عملي:

مثال عملي باستخدام B_I:



توضيح للشرح:



VB0

1	1	0	1	0	1	0	1

VW6

0	0	0	0	0	0	0	0	1	1	0	1	0	1	0	1

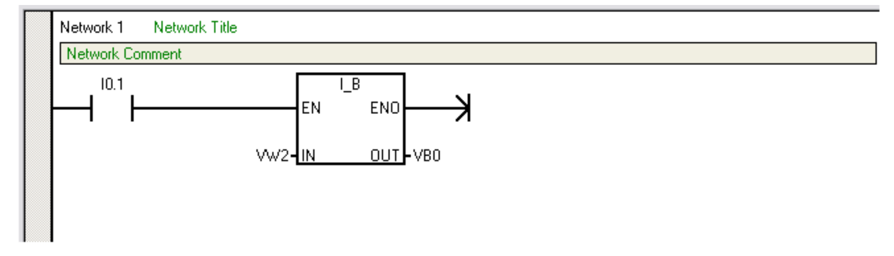
لم تتغير القيمة ولكن تم تغير الحجم من Byte إلى Word.

ملاحظة:

كل ما يكتب على byte يمكن أن ينقل على word لأننا ننقل من الحجم الأصغر إلى الحجم الأكبر ولكن العكس ليس بمضمون لأنه يعتمد على القيمة الفعلية المسجلة داخل الذاكرة.

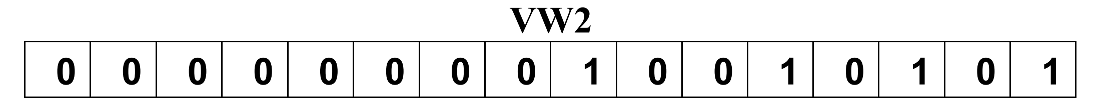
مثال عملي:

مثال عملي باستخدام I_B:



توضيح للشرح:

VW2



0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	1

VB0

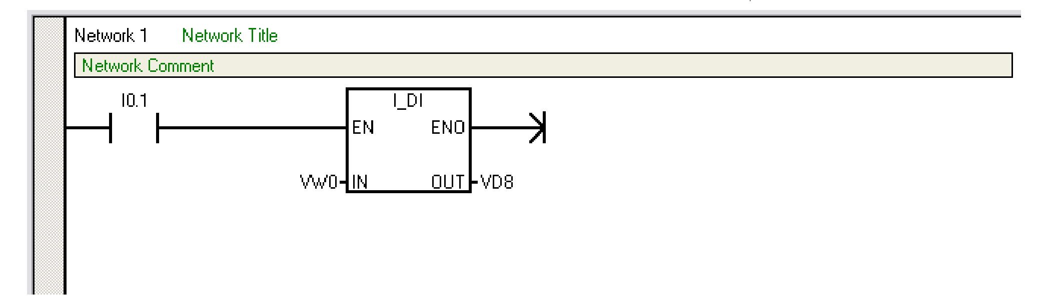
1	0	0	1	0	1	0	1

لم تتغير القيمة ولكن تم تغير الحجم من Word إلى Byte.

يجب مراعاة أن لا يكون محتوى الـ word أكبر من أن يكتب على byte.

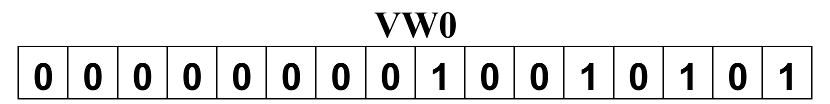
مثال عملي:

مثال عملي باستخدام I_D:



توضيح للشرح:

VW0



0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	1

VD8

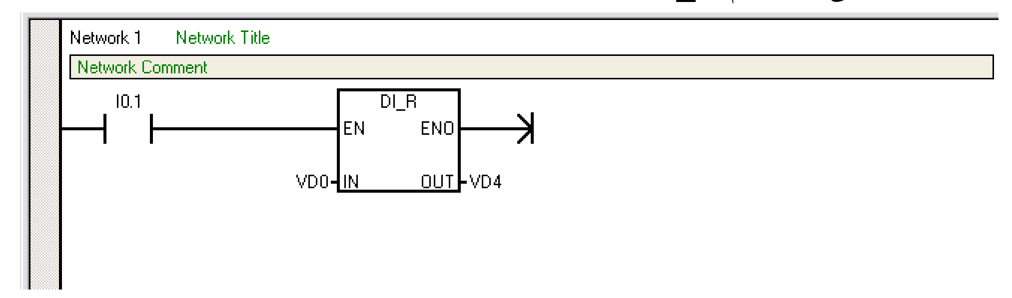
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	1

لم تتغير القيمة ولكن تم تغير الحجم من Word إلى Dword.

كل ما يكتب على Word يمكن أن ينقل على Dword لأننا ننقل من الحجم الأصغر إلى الحجم الأكبر.

مثال عملي:

مثال عملي باستخدام DI_R:



توضيح للشرح:

VD0

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

VD4

0	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

لم تتغير القيمة ولم يتغير الحجم ولكن تم تغير النظام من أرقام صحيحة لأرقام عشرية.

مثال عملي:

مثال عملي باستخدام ROUND:

Network 1 Network Title

Network Comment

I0.1 ROUND EN ENO

VD0 IN OUT VD4

توضيح للشرح:

VD0

0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1

VD4

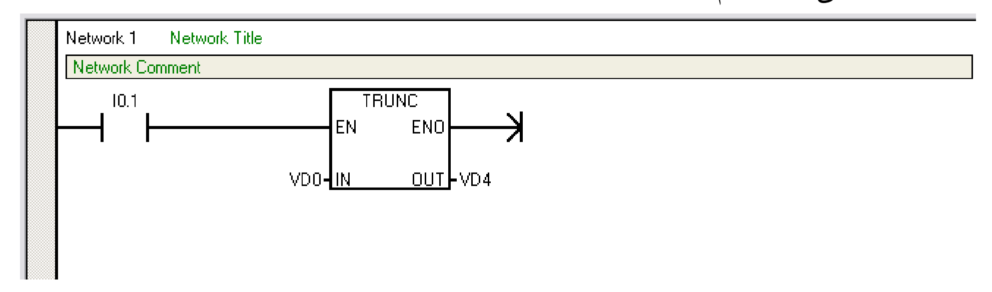
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

أمر ROUND يقرب إلى أقرب رقم صحيح.

الرقم المكتوب فى VD0 هو ٦,٤ بينما المكتوب فى VD4 هو ٦

مثال عملى:

مثال عملى باستخدام TRUNC:



توضيح للشرح:

VD0

0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1

VD4

0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

أمر TRUNC يمسح الرقم المكتوب بعد العلامة و يكتب الرقم الصحيح فقط.

الرقم المكتوب فى VD0 هو ٦,٤ بينما المكتوب فى VD4 ٤

ما الفائدة من استخدام المحولات؟

١– فى حالة استخدام قيمة من على IB0 للاستخدام مع مؤقت زمنى أو مع عداد فسوف نلاحظ أنه توجد مشكلة وهى أن القيمة الخاصة بالمفاتيح هى على حجم byte بينما المؤقت الزمنى أو العداد يعمل كل منهم على ذاكرة بحجم word ولذالك فأنة يتم استخدام المحولات من byte إلى word وذالك عن طريق العملية B-I.

٢– فى حالة استخدام قيمة من على مؤقت زمنى أو من على عداد للاستخدام مع قيمة أخرى (معادلة رياضية) تحتوى على أرقام عشرية فسوف نلاحظ أنه توجد مشكلة وهى أن القيمة الخاصة بالمؤقت الزمنى أو بالعداد هى بحجم word بينما الأرقام العشرية أى الأرقام غير الصحيحة تكتب على ذاكرة بحجم Dword ولذالك فأنة يتم استخدام:

- **أولاً**: المحولات من word إلى Dword وذالك عن طريق العملية I-DI لتغيير الحجم مع الاحتفاظ بنفس القيمة .
- **ثانياً**: المحولات من Dword إلى Real وذالك عن طريق العملية DI-R لتغيير القيمة الصحيحة لقيمة عشرية دون تغيير الحجم.

٣– فى حالة استخدام قيمة عشرية بحجم Dword فمن المؤكد أن القيمة لن تكون صحيحة بل ستحتوى على أى رقم بعد العلامة العشرية حتى وأن كان هذا الرقم هو صفر وبينما فى حالة التعامل مع عداد ستكون المشكلة ليست فى الحجم فقط بل فى الـ format أيضاً فلذالك سيستخدم:

- **أولاً**: أمر TRUNC لحذف أى أرقام بعد العلامة العشرية دون التغير فى الحجم.
- **ثانياً**: المحولات من Dword إلى Word وذالك عن طريق العملية DI-I لتغيير الحجم دون تغير القيمة.

الباب الثانى عشر

الترحيل و الدوران

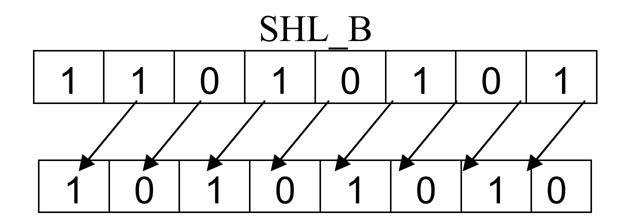
- شرح الترحيـــــــــل SHIFT.
- الأحجام المختلفة للترحيـــــــــل.
- شرح الـــــدوران ROTATE.
- الأحجام المختلفة للــــــــدوران.
- الفرق بين الترحيـــل يميناً ويساراً.
- الفرق بين الـــدوران يميناً ويساراً.
- الأخـــــطاء الممكن التعرض لها.
- تمـــــــــارين عملية للتوضيح.

الترحيل و الدوران:

الترحيل:

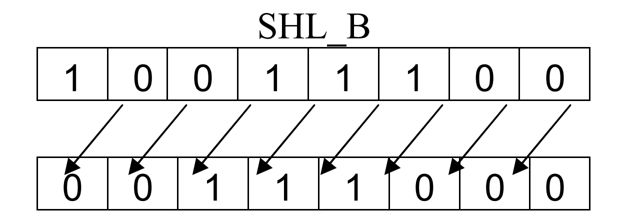
يقوم بترحيل محتويات الـ bits لليمين أو لليسار حسب النوع, بحيث أن محتوى الـ bits التى تخرج عن حدود الذاكرة تمسح تلقائياً.

مثال:



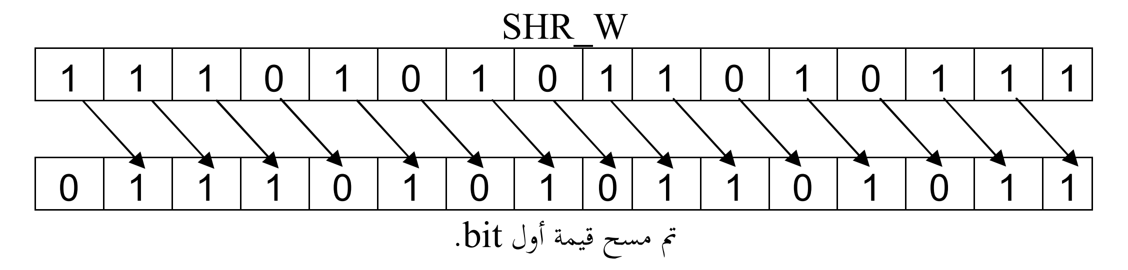
حيث تم مسح قيمة أخر bit.

مثال أخر:



حيث تم مسح قيمة أخر bit.

مثال:



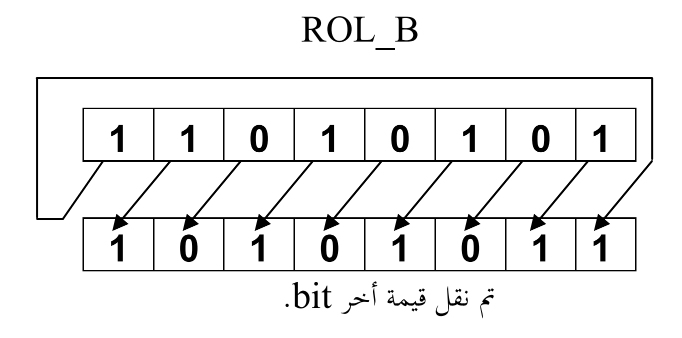
تم مسح قيمة أول bit.

الدوران:

يقوم بدوران محتويات الـ bits لليمين أو لليسار حسب النوع, ولكن محتوى الـ bits التى تخرج من الحجم الذى نتعامل معه لا تمسح بل تعود لنفس الذاكرة من الناحية الأخرى.

مثال:

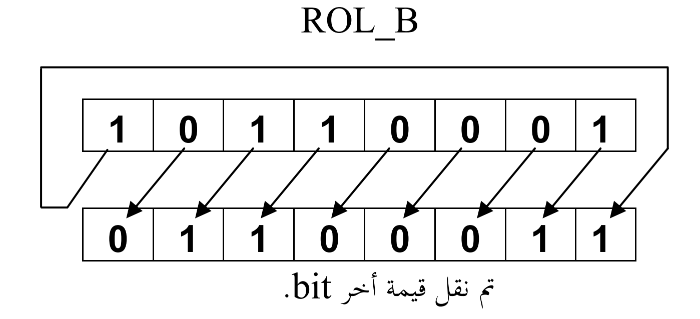
ROL_B



تم نقل قيمة أخر bit.

مثال أخر:

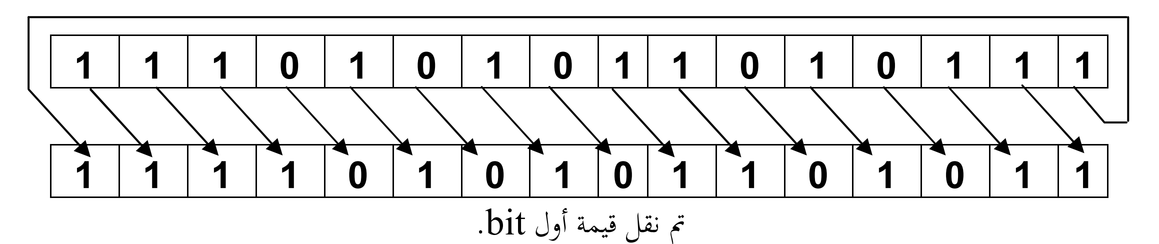
ROL_B



تم نقل قيمة أخر bit.

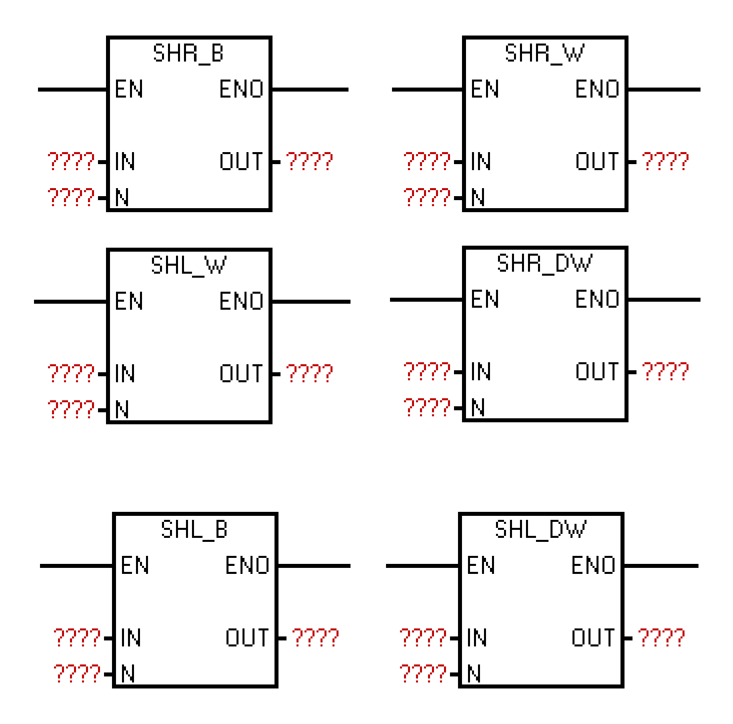
مثال:

ROR_W

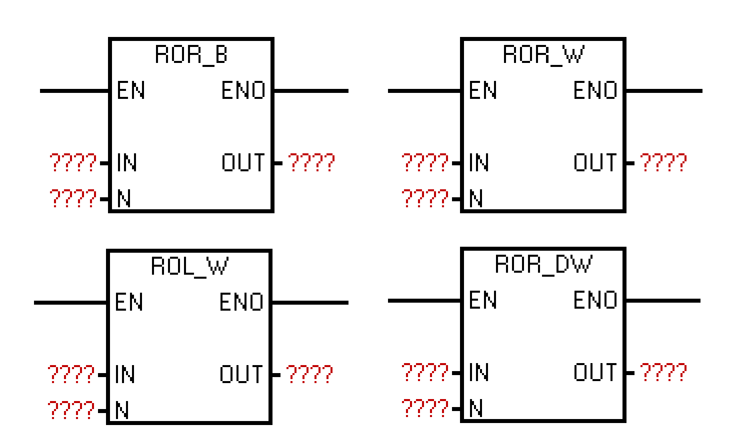


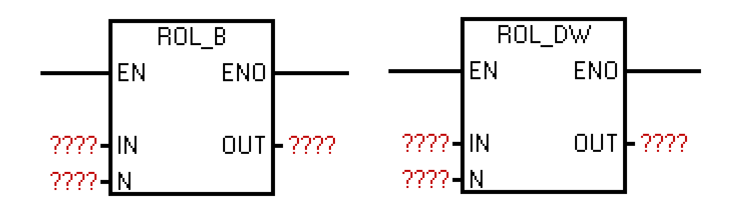
تم نقل قيمة أول bit.

الترحيل:



الدوران





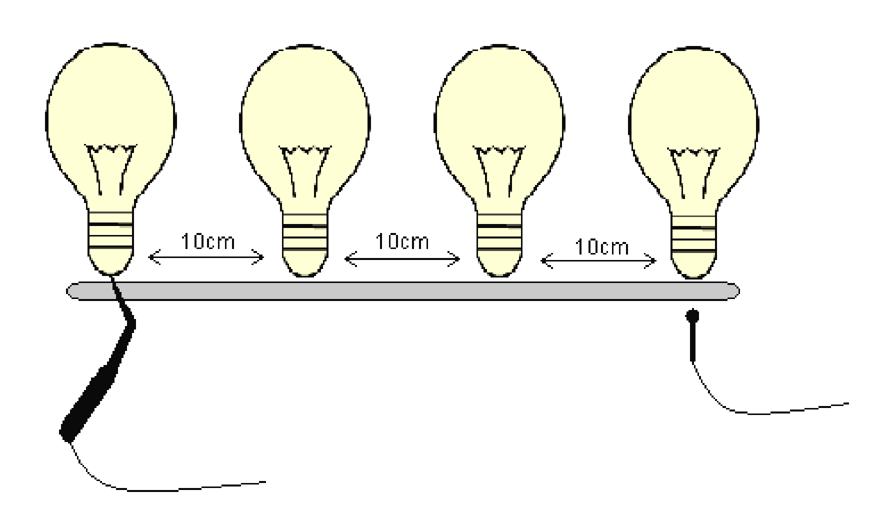
شرح أنواع الترحيل والدوران

م	الأسم	الشرح	الشكل
١	SHR_B	يقوم SHR_B بترحيل القيم الخاصة بال BITS داخل ال BYTE من اليسار إلى اليمين .	SHR_B: EN ENO ????-IN OUT-???? ????-N
٢	SHR_W	يقوم SHR_W بترحيل القيم الخاصة بال BITS داخل ال WORD من اليسار إلى اليمين .	SHR_W: EN ENO ????-IN OUT-???? ????-N
٣	SHR_DW	يقوم SHR_DW بترحيل القيم الخاصة بال BITS داخل ال DWORD من اليسار إلى اليمين .	SHR_DW: EN ENO ????-IN OUT-???? ????-N
٤	SHL_B	يقوم SHL_B بترحيل القيم الخاصة بال BITS داخل ال BYTE من اليمين إلى اليسار.	SHL_B: EN ENO ????-IN OUT-???? ????-N

٥	SHL_W	يقوم SHL_W بترحيل القيم الخاصة بال BITS داخل ال WORD من اليمين إلى اليسار.	SHL_W EN ENO ????-IN OUT-???? ????-N
٦	SHL_DW	يقوم SHL_DW بترحيل القيم الخاصة بال BITS داخل ال DWORD من اليمين إلى اليسار.	SHL_DW EN ENO ????-IN OUT-???? ????-N
٧	ROR_B	يقوم ROR_B بدوران القيم الخاصة بال BITS داخل ال BYTE من اليسار إلى اليمين .	ROR_B EN ENO ????-IN OUT-???? ????-N
٨	ROR_W	يقوم ROR_W بدوران القيم الخاصة بال BITS داخل ال WORD من اليسار إلى اليمين .	ROR_W EN ENO ????-IN OUT-???? ????-N
٩	ROR_DW	يقوم ROR_DW بدوران القيم الخاصة بال BITS داخل ال DWORD من اليمين إلى اليسار.	ROR_DW EN ENO ????-IN OUT-???? ????-N

١٠	ROL_B	يقوم ROL_B بدوران القيم الخاصة بال BITS داخل ال BYTE من اليمين إلى اليسار.	ROL_B: EN ENO, ????-IN OUT-????, ????-N
١١	ROL_W	يقوم ROL_W بدوران القيم الخاصة بال BITS داخل ال WORD من اليمين إلى اليسار.	ROL_W: EN ENO, ????-IN OUT-????, ????-N
١٢	ROL_DW	يقوم ROL_DW بدوران القيم الخاصة بال BITS داخل ال DWORD من اليمين إلى اليسار.	ROL_DW: EN ENO, ????-IN OUT-????, ????-N

رسم توضيحى للتمرين العملى:

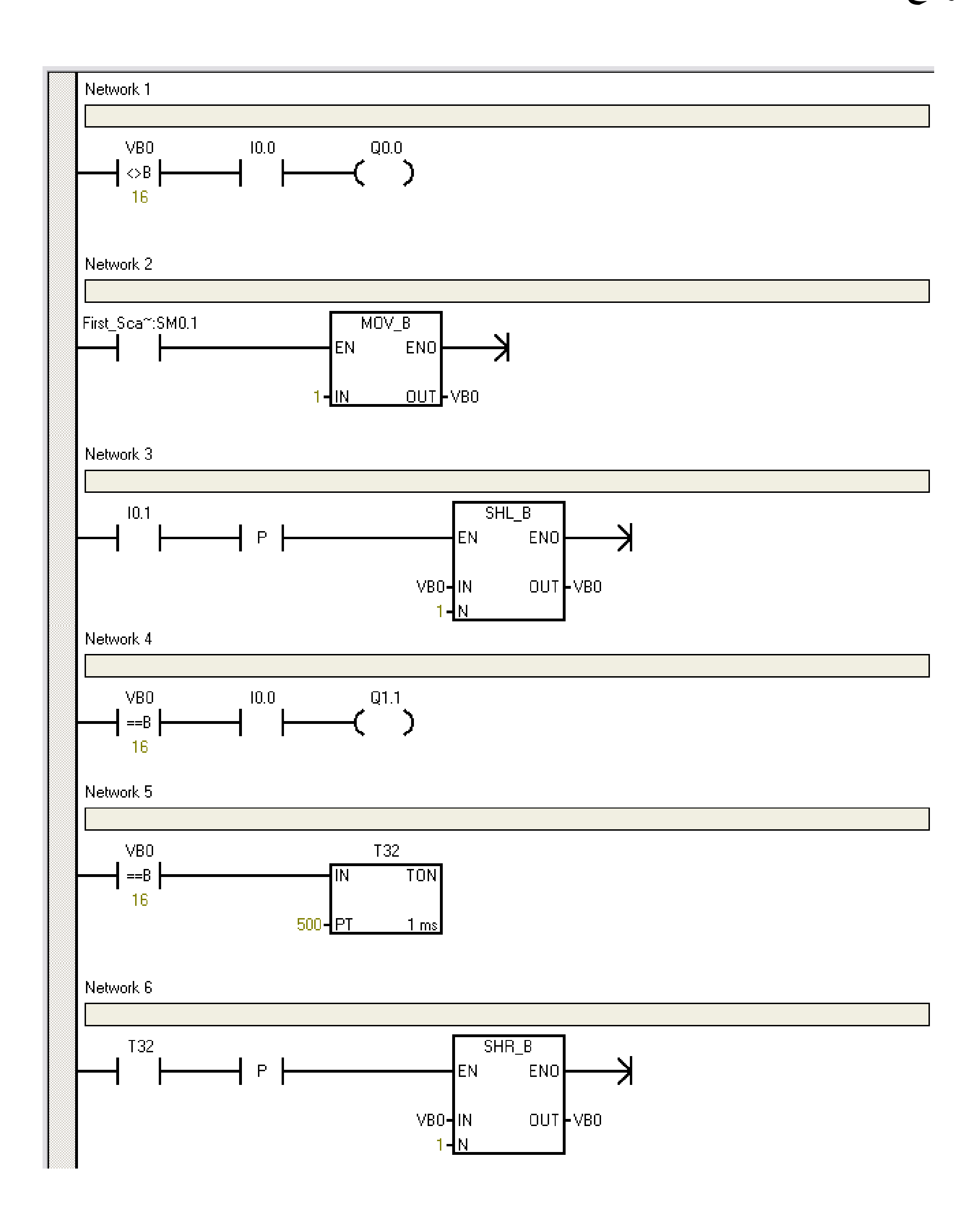


التمرين عملى:

مكينة تقوم بلحام الأطراف الداخلية المثبتة فى القعدة الخاصة باللمبة و لكن نظراً لشدة درجة الحرارة بالقرب من مكينة اللحام سوف يتم و ضع الحساس بعيداً عن مكان مكينة اللحام, المسافة بين اللمبات متساوية وهى تعتمد على حركة السير فهو يتحرك نصف ثانية ويقف نصف ثانية.

عدد الدخل	نوع الدخل	أسم الدخل
١	n.c.	I0.0/S1
٢	n.o.	I0.1/S2
عدد المؤقتات الزمنية	**نوع المؤقتات الزمنية**	**أسم المؤقتات الزمنية**
١	TON	T32
عدد المتغيرات	**نوع المتغيرات**	**أسم المتغيرات**
١	Byte	VB0
عدد الترحيل	**نوع الترحيل**	**أسم الترحيل**
١	SHL_B	VB0
٢	SHR_B	VB0
عدد مفاتيح المقارنة	**نوع مفاتيح المقارنة**	**أسم مفاتيح المقارنة**
١	==B	VB0
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتكتور	Q0.0/K1M
٢	كونتكتور	Q1.1/K2M

البرنامج:


Network 1
VB0
<>B
16
I0.0
Q0.0
Network 2
First_Sca~:SM0.1
MOV_B
EN
ENO
1
IN
OUT
VB0
Network 3
I0.1
P
SHL_B
EN
ENO
VB0
IN
OUT
VB0
1
N
Network 4
VB0
==B
16
I0.0
Q1.1
Network 5
VB0
==B
16
T32
IN
TON
500
PT
1 ms
Network 6
T32
P
SHR_B
EN
ENO
VB0
IN
OUT
VB0
1
N

الشرح:

:Network1

سوف يعمل السير طالما قيمة الـ VB0 مختلفة عن ١٦.

:Network2

فى أول Cycle سوف تكون قيمة VB0 هى ١ أى أن قيمة V0.0 هي ١.

:Network3

كل مرة تمر اللمبة أمام الحساس سوف يتم تطبيق مبدأ ترحيل الـ bits الخاصة بالـ Byte VB0 لليسار.

:Network4

عندما تصبح قيمة الـ VB0 هى ١٦ هذا يعنى أنه توجد الآن لمبة أمام مكينة اللحام.

:Network5

سوف يعمل المؤقت الزمنى لكى يقوم بفصل مكينة اللحام تلقائياً.

:Network6

فيقوم المؤقت الزمنى بترحيل الـ bits الخاصة بالـ Byte VB0 لليمين فيعمل السير الذى يحمل اللمبات مرة أخرى, وهكذا.

الباب الثالث عشر

العلامات

- شرح العـــــــــــــــــــلامات.
- الهدف من استخدام العــــــلامات.
- كيفية التنقل بين الأفــــــرع بسرعة.
- المفاتيح المستخدمة مع العــلامات.
- تمـــــــــــــــارين عملية للتوضيح.

العلامات:

تستخدم العلامات لتميز الأفرع عن بعضها و للتنقل من فرع إلى أخر بسهولة خاصة فى البرامج الكبيرة عندما تكون هناك فروع برمجة مرتبطة ببعضها و فى نفس الوقت تبعد عن بعضها من حيث تواجدها فى البرنامج, قد يحدث فى أى برنامج إلى أن نضطر إلى تغير ترتيب أماكن أفرع البرمجة حتى يعمل البرنامج بصورة صحيحة ولكن حين يتطلب الموضوع إلى العودة فى المستقبل للعمل على نفس البرنامج مرة أخرى نكون لم نعد نتذكر بعد أى من أفرع البرمجة مرتبطة ببعضها وهذا يتطلب وقت كبير لإعادة قراءة البرنامج بالكامل من البداية, الأمر الذى كان يمكن حله من البداية فى حاله تم استخدام العلامات كما سنوضح الآن.

المفاتيح المستخدمة:

١- مفتاح وضع العلامات Toggle bookmark

حيث يتم الوقوف على الفرع المراد ثم الضغط على المفتاح الخاص بوضع العلامات.

٢- مفتاح للتنقل للعلامات التالية Next bookmark

حيث يتم التنقل للإمام مباشراً إلى الفرع المحدد دون ضرورة المرور على باقى الأفرع.

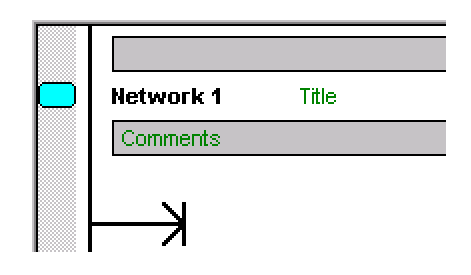
٣- مفتاح للتنقل للعلامات السابقة Previous bookmark

حيث يتم التنقل للخلف مباشراً إلى الفرع المحدد دون ضرورة المرور على باقى الأفرع.

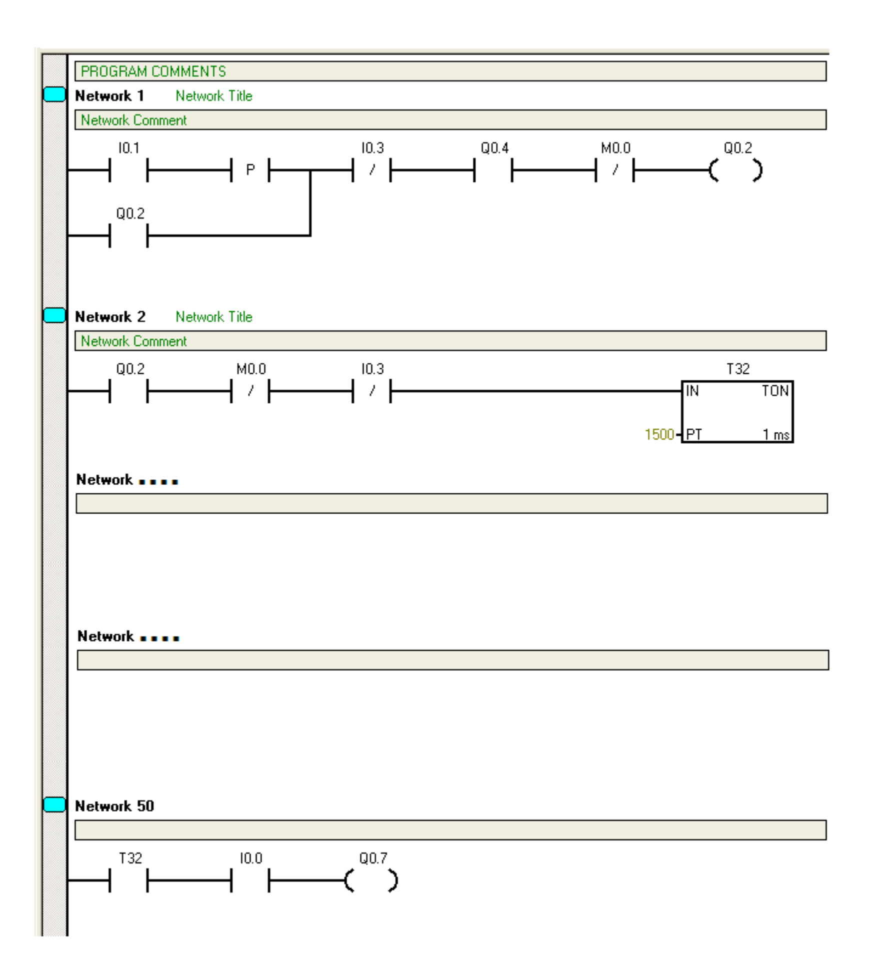
٤- مفتاح حذف جميع العلامات Remove all bookmarks

حيث يتم حذف جميع العلامات من جميع الأفرع الموجودة بالبرنامج.

الشكل العام للعلامات, لل Bookmarks:



البرنامج:



توضيح:

ليس الفكرة فى البرنامج بل الفكرة انه يمكن أن يتم التنقل من الفرع الثانى إلى الفرع الخمسين فى الحال, فبالرغم من أن الفرع الخمسين يحتوى على الخرج الذى يعمل بواسطة المؤقت الزمنى المتواجد فى الفرع الثانى ولكن لظروف ذات علاقة بالأولويات لم نتمكن من وضعهما بالقرب من بعضهما ولكن تم حل المشكلة باستخدام العلامات Bookmarks.

يتم وضع هذه العلامة بترتيب عند الأفرع المرتبطة ببعضها.

الفهرس

الفهرس

الفهرس

الفهرس

الفهرس

الكتب التي صدرت عن معهد السالزيان الإيطالى "دون بوسكو"

🕮 **محركات, مولدات و محولات التيار المتردد**	وجيه جرجس
🕮 **دوائر التحكم الآلي** الجزء الأول	وجيه جرجس
🕮 **دوائر التحكم الآلي** الجزء الثاني	وجيه جرجس
🕮 **الغسالة الفول أوتوماتك** الجزء الأول	وجيه جرجس
🕮 **الغسالة الفول أوتوماتك** الجزء الثاني	وجيه جرجس
🕮 **الدوائر العملية للضغوط الهوائية و الكهروهوائية**	وجيه جرجس
🕮 **غسالة الأطباق**	وجيه جرجس
🕮 **زانوسى الموديلات القديمة 14-16-18بروجرام**	وجيه جرجس
🕮 **موديلات الغسالة كريازى**	وجيه جرجس
🕮 **الدوائر الكهربائية للتركيبات المنزلية**	نبيل رزق
🕮 **صيانة وإصلاح الأجهزة المنزلية**	نبيل رزق
🕮 **أفكار التكيف و التبريد للدوائر الميكانيكية**	إميل فتح الله
🕮 **أفكار التكيف و التبريد للدوائر الكهربائية**	إميل فتح الله
🕮 **أفكار التكيف و التبريد الخدمة والأعطال**	إميل فتح الله
🕮 **برمجة التحكم المنطقي .P.L.C** الجزء الأول	ريمون كمال
🕮 **برمجة التحكم المنطقي .P.L.C** الجزء الثاني	ريمون كمال
🕮 **برمجة التحكم المنطقي .P.L.C** إعطال و تمارين عملية	تحت التحضير

برمجة التحكم المنطقى

P.L.C.

أعطال – صيانة – تمارين

إعداد

ريمون كمال

معهد السالزيان الإيطالى "**دون بوسكو**"

٢ شارع عبد القادر طه – الساحل ت: ٢٤٥٧٩٦٥٠ – ٢٤٥٧٦٧٩٤

معهد فنى – معهد صناعى

دورات تدريبية سريعة مركزة

دورات تدريبية تعليمية للمدرسين

المراجع

١. أجهزة التحكم المبرمج وتطبيقاتها العملي.

1. Controllore a logica programmabile P. Bani.
2. Siemens Programmable Controller Manual.

طبعة جديدة

2012

أسم الكتاب: **برمجة التحكم المنطقى P.L.C.**

أعطال – صيانة – تمارين

طباعة

رقم الإيداع:

الترقيم الدولى:

شكر و إهداء

أهدى هذا الكتاب إلى أبى وأمي الذين لهم كل الفضل بأن أعمل في هذا المجال وهم الذين شجعوني على عمل هذا الكتاب بكل جهد وإخلاص شاكر الله و إياهم وكل من ساهم في تقديمه.

وأشكر أيضاً كل المعلمين الأفاضل الذين ساعدوا على خروج هذا الكتاب إلى المليء.

- ☜ **المدير الإيطالي للمعهد**: الأب رينسو ليوناردوسى
- ☜ **الناظر السابق للمعهد**: الأب بيرناردو أشيربونى
- ☜ **مدير الدورات التدريبية**: أ. ماجد چورچ
- ☜ **أستاذ التحكم**: أ. نبيل رزق **–** أ. وجيه جرجس – أ. إلبير صالح
- ☜ **أستاذ التكيف والتبريد**: أ. إميل فتح الله
- ☜ **أستاذ الـ PLC**: أ. ماجد موريس **–** أ. ماجد عريان **–** أ. چيوليو جالو **–** أ. محسن أنطون

أشكر كل من أرسلوا لى التعليقات بخصوص الجزء الأول والثانى من الكتاب وقد حاولت قدر المستطاع تلبية متطلباتهم فى هذا الكتاب وأتشرف باستقبال المزيد من تعليقات السادة القراء بخصوص هذا الكتاب على عنوان البريد التالى

plcbook@hotmail.com

مقدمة

نظراً للتقدم العلمى السريع المرتبط بالمجال الصناعى وخاصة من الناحية الكهربية أصبح لا غنى عن الربط بين عالم الصناعة وبين التكنولوچيا العصرية ويتمثل هذا الربط بواسطة استخدام أجهزة التحكم المنطقى بمختلف أنوعها والتى تستحق أن تسمى بالأجهزة الذكية نظراً لما تقدمه فى المجال الصناعى من: سهولة فى تصميم البرامج، ومرونة فى أكتشاف الأعطال، ومساعدة فى حل المشاكل، ...الخ

تنقسم معرفة أجهزة التحكم المنطقى إلى أمور عديدة من أهمها:

تصميم برامج – اكتشاف أعطال – حل مشاكل

قد تم التركيز بشكل كبير فى الجزء الأول من كتاب برمجة التحكم المنطقى *P.L.C.* على معرفة أساسات تصميم البرامج بطريقة سلسة وباستخدام أسهل لغات البرمجة وفى الجزء الثانى من كتاب برمجة التحكم المنطقى *P.L.C.* تم استكمال شرح تصميم البرامج باستخدام أساليب ذات أكثر تطور وأقدم لكم الآن كتاب خاص بالأعطال وبحلول المشاكل وبتصميم التمارين والأفكار العملية.

يحتوى هذا الكتاب على نسبه كبيرة جداً من الأعطال التى يمكن التعرض لها أثناء العمل اليومى بل ويحتوى الكتاب أيضاً على الحلول الممكنة لكل المشاكل وقد تم تخصيص فصل بالكامل لبعض الأفكار العملية والدوائر المنطقية.

و خاصاً لفهم التمارين لا يشترط فقط القراءة بترتيب بل يجب أيضاً أن تربط كل شرح و كل رمز بالرسم الموجود ولا تقوم بالقراءة بطريقة عابرة خاصاً لأن هذا الكتاب يحتوى على عدد كبير من الأفكار العملية المختلفة, حيث تم شرح البرمجة بطريقة عامة دون اللجوء إلى ماركة بعينها وهذا لكى يخدم كل من يعمل مع وحدات التحكم المنطقى بمختلف أنواعها.

لذالك أقدم لكم هذا الكتاب لخدمة كل من يدرس أو يعمل فى هذا المجال وأتمنى من الله أن يجد كل من يقرأ هذا الكتاب نفعاً له.

المؤلف

الباب الأول

الأعطال

- **مقدمة عن الأعطـــــــــــــــــــال.**
- **كيفية أكتشاف مكان العطـــــــــل تلقائياً.**
- **كيفية حل الأعطـــــــــــــــال بسهولة.**
- **نسبة الأعطـال داخل وخارج وحدة البرمجة.**
- **الصفح المستخدمة لكشف الأعطـــــال.**
- **أعطال المدخـــــــــــــــلات وأسبابها.**
- **أعطال المخـــــــــــــــرجات وأسبابها.**
- **بعض أنواع الأعطـــــــــــــال المتدولة.**
- **تمارين عملية لتحديد سبب ومكان العطـــل.**

الأعطال:

دائماً ما يهتم كل من يدرس فى هذا المجال بأن يتعرف جيداً على كيفية اكتشاف وحل العطل بل وتجنبها من الأساس أيضا ولكنى تعمدت أن لا أتكلم عن هذا الفصل فى الجزء الأول والثانى من الكتاب لسببين, الأول حتى لا ينتاب القارئ البسيط أو المستجد شعور بالصعوبة وتعقد الأمر بسبب وجود أعطال تقع فيها وحدة البرمجة وهو الشيء الذى كان يجعل البعض يستصعب دوائر التحكم, إما السبب الثانى فهو حتى أن يكون قد تمكن القارئ من البرمجة بصورة جيدة لكى تتيح له الفرصة بعد ذالك للتمكن من اكتشاف وحل الأعطال بسهوله بل وتجنبها من الأساس.

بداية للتفاؤل فأن نسبة الأعطال الخاصة بوحدة البرمجة PLC مقارنتاً بدوائر التحكم لا تقارن لأن أعطال جهاز البرمجة هى ضئيلة و محدودة جداً بل وتساعد وحدة البرمجة ذاتها فى اكتشاف وحل بعض الأعطال ولكن هذا يعتمد على مدى قوة هذا العطل.

أهداف الفصل:

- طريقة منهجية لإصلاح الأعطال.
- إصلاح الأعطال خارج الـ PLC.
- المشاكل في الـ PLC التي تنتج عن الأخطاء القاتلة و الأخطاء غير القاتلة بما فيها الأخطاء الحسابية أو الأخطاء في البرنامج.
- بعض التعليمات و التقنيات البرمجية التي يمكن أن تساعد في الكشف عن التصرفات غير المرغوبة.

الطريقة المنهجية:

المشاكل المتعلقة بوحدة البرمجة يجب أن تحل باستخدام الطرق التالية:

١- حدد المشكلة.

٢- قرر ما هو التصرف الصحيح الذي يجب أن يسلكه النظام.

٣- طبق الحل الأفضل (سنساعدك فى ذالك خلال هذا الفصل).
٤- تأكد أن المشكلة قد تم حلها وأن وحدة البرمجة تعمل بنظام صحيح.

ملاحظة:

- ليس الحل الأسهل هو الأفضل دائماً وخاصاً في الأنظمة الصناعية المعقدة المستخدمة هذه الأيام. تستطيع وحدات البرمجة مساعدتك في تحديد ما هي المشكلة إذا كانت المشكلة فى الأساس بسبب وحدة البرمجة ولكن كن مستعداً دائماً للبحث عن الحل خارج وحدة البرمجة لأن أغلبية المشاكل والأعطال بكون خارج نطاق البرمجة.
- يحدث في كثير من المنشآت الصناعية الكثير من المواقف الغريبة عن وحدات برمجة ممتازة تم تغييرها ظاناً أنها السبب فى المشكلة بينما قد تكون المشاكل ناتجة عن أعطال رولمانات أو أسلاك كهربائية مقطوعة بل وربما بسبب أشياء لا تخطر على عقلنا.

ببساطة يتبع الشخص العادى الخطوات التالية:

استخدم مميزات كشف الأعطال الموجودة في وحدة البرمجة لتحديد ما هي المشكلة.
البحث عن حلول ممكنة بتفقد التوصيلات خارج وحدة البرمجة.

ملاحظة:

- عادة ما تحتوي البرامج التى يكتبها المبرمجون الجدد على أخطاء، ولكن أيضاً فإن نظام تحكم بني حديثاً قد يحتوي على عناصر معطوبة أو وصلات خاطئة أيضاً.

الفرق بين الأعطال والمشاكل:

هناك فرق كبير بين الأعطال errors والمشاكل problems, فالأعطال هى ما تحدث داخل وحدة البرمجة إما المشاكل هى ما تحدث خارج وحدة البرمجة وتقريباً ما تكون نسبة الأعطال داخل وحدة البرمجة إلى المشاكل خارج وحدة البرمجة هى نسبة ضئيلة جداً فهى نسبة ٥% إلى نسبة ٩٥%

المشاكل خارج وحدة البرمجة: Problems

فبالنسبة للمشاكل المتواجدة خارج وحدة البرمجة قد تكون خاصة بالمدخلات أو المخرجات ويمكن بسهوله تحديد سبب ومكان المشكلة خارج وحدة البرمجة بالاستعانة بصفحة جدول الحالات status chart التى تم شرحها فيما سبق.

فى حاله ملاحظة أى اختلاف فى طريقة التشغيل أو خلل معين يتم أولاً البحث عن مشاكل خارج وحدة البرمجة ثم بعد ذالك البحث عن أعطال داخل وحدة البرمجة وذالك لأننا نبحث عن الأعطال فى الجزء السهل أولاً ومن ثم نبحث فى الجزء الصعب, الخطوات المستخدمة لاكتشاف المشاكل خارج جهاز الـ PLC هى:

أولاً:

تتم كتابة جميع المداخل والمخرج بصفحة status chart كما هو موضح بالشكل التالى.

	Address	Format	Current Value	New Value
1	I0.0	Bit		
2	I0.1	Bit		
3	I0.2	Bit		
4	Q0.0	Bit		
5	Q0.1	Bit		
6	Q0.2	Bit		

ثانياً:

يتم الضغط على مفتاح chart status لعرض الحالة الخاصة بجميع المداخل والمخارج بصفحة status chart كما هو موضح بالشكل التالى.

	Address	Format	Current Value	New Value
1	I0.0	Bit	2#0	
2	I0.1	Bit	2#1	
3	I0.2	Bit	2#1	
4	Q0.0	Bit	2#0	
5	Q0.1	Bit	2#0	
6	Q0.2	Bit	2#1	

أعطال المدخلات:

بيان لبعض الأعطال و الأسباب والحلول, للتحقق من المشاكل الخاصة بالمداخل:

إذا كان المفتاح المفتوح بالخارج يشار إليه بالداخل بأنه مفتوح أيضا والمفتاح المغلق بالخارج يشار إليه بالداخل بأنه المغلق أيضا فهذا يدل على أنه لا توجد أى مشكلة بالنسبه للمداخل, إما إذا كان المفتاح مغلق بالخارج بينما يشير جهاز الـ PLC بأن المفتاح مفتوح بالداخل فهذا بكل بساطة يشير إلى أنه توجد مشكله بهذا المفتاح (لا توجد كهرباء على طرف المفتاح – الطرف المشترك على الوحدة غير موصل – تم التوصيل على العزل – خطاء فى توصيل وحدة البرمجة نفسها) يمكنك النظر إلى طريقة توصيل مختلف أنواع أجهزة الـ PLC فى الجزء الأول من هذا الكتاب.

توجد طريقتان لمعرفة سبب العطل وحل العطل فى حاله أن المفتاح مغلق بالفعل ولكن يشار إلى أنه ليس مغلق بل مفتوح أو أن كان المفتاح مفتوح بالفعل ولكن يشار إلى أنه ليس مفتوح بل مغلق .

أ -**الطريقة الأولى**:

يجب تتبع الخطوات حسب الجدولين التاليين حتى التوصل إلى سبب العطل ومن ثم التوصل إلى حل العطل.

الجدول الأول (حينما يشار إلى أن المفتاح مفتوح بينما هو مغلق بالفعل.)

فى حاله وجود أى اختلاف بين الحالة الحقيقية للمدخلات وبين الحالة المبينة فى صفحة status chart فهذا يدل على وجود مشكلة,فمثلاً إذا كان الشكل كالتالى:

I0.1
PLC

أن كان مفتاح I0.1 مغلق بالخارج كما بالشكل بينما تشير وحدة البرمجة إلى أن المفتاح غير مغلق فهذا يدل بالتأكيد على أن المشكلة هى فى المفتاح I0.1, للتعرف على السبب الأساسى للمشكلة تتبع الخطوات التالية.

م	السبب	الحل
١	مشكلة بالمفتاح نفسه	- يجب التأكد من سلامة المفتاح باستخدام الأوم ميتر. - يجب تغير المفتاح.

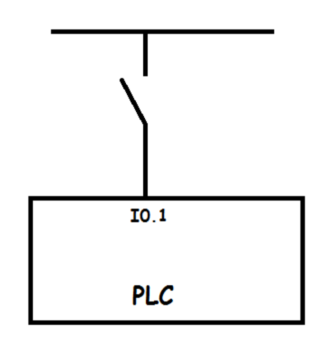
٢	مشكلة بالسلك نفسه	– يجب التأكد من عدم وجود انقطاع بالسلك. – يجب تغير السلك.
٣	مشكلة فى التلامس	– يجب إعادة ربط الطرف الموصل على الـ PLC. – يجب إعادة ربط الطرف الموصل بمصدر الكهرباء.
٤	مشكلة فى قيمة الجهد	– يجب التأكد من أن القيمة لا تقل عن ٢٤ فولت. – يمكن قياس قيمة الفولت باستخدام الفولت ميتر.
٥	مشكلة فى تغذية وحدة البرمجة	– يجب التأكد من أن مصدر الكهرباء (power supply) يعمل بطريقة صحيحة. – التأكد من سلامة الفيوز الخاص بالـ power supply.
٦	مشكلة فى توصيل وحدة البرمجة	– التأكد من توصيل الطرف المشترك الموجب أو السالب حسب نوع الـ PLC, أنظر صفحة ٣٧ بالجزء الأول من كتاب التحكم المنطقى. – التأكد من توصيل طرف المفتاح بطريقة سليمة على وحدة البرمجة. – التأكد من أنه لم يتم ربط الطرف الموصل على العزل.
٧	مشكلة فى وحدة البرمجة	– التأكد من عدم تطبيق أمر force على الدخل, أنظر صفحة ٥٠ بالجزء الثانى من كتاب التحكم المنطقى. – يجب الضغط على read all forced لمعرفة إذا كان هناك أمر force تم تطبيقه من قبل على أى مفتاح.

٨	مشكلة فى وحدة البرمجة	– التأكد من عدم وجود عطل فى وحدة المدخلات. – التأكد من أن لمبة diagnostic الخاصة بأعطال وحدة المدخلات لا تضيء (أى لا تشير إلى وجود أعطال).

الجدول الثانى (يشار إلى أن المفتاح مغلق بينما هو مفتوح بالفعل.)

حاله وجود أى اختلاف بين الحالة الحقيقية للمدخلات وبين الحالة المبينة فى صفحة status chart فهذا يدل على وجود مشكلة,فمثلاً إذا كان الشكل كالتالى:

أن كان مفتاح I0.1 مفتوح بالخارج كما بالشكل بينما تشير وحدة البرمجة إلى أن المفتاح مغلق فهذا يدل بالتأكيد على أن المشكلة هى فى المفتاح I0.1, للتعرف على السبب الأساسى للمشكلة تتبع الخطوات التالية.



م	السبب	الحل
١	مشكلة بالمفتاح نفسه	يجب تغيير المفتاح
٢	مشكلة فى التوصيل	قد تحدث فى حاله توصيل أكثر من مفتاح على نفس النقطة.
٣	مشكلة فى توصيل وحدة البرمجة	التأكد من توصيل الطرف المشترك الموجد أو السالب حسب نوع الـ PLC, أنظر صفحة ٣٧ بالجزء الأول من كتاب التحكم المنطقى.

٤	مشكلة فى وحدة البرمجة	التأكد من عدم تطبيق أمر force على الدخل, أنظر صفحة ٥٠ بالجزء الثانى من كتاب التحكم المنطقى.
٥	مشكلة فى وحدة البرمجة	التأكد من عدم وجود عطل فى وحدة المدخلات.

ب –الطريقة الثانية:

هى طريقة تستخدم لكى يسهل التوصل إلى معرفة إذا كان سبب العطل هو بسبب مشكلة خارج أم داخل وحدة البرمجة, يجب تتبع الخطوات التالية لمعرفة السبب الحقيقى للعطل.

١– يكتب عنوان الدخل المشكوك فيه داخل الـ address فى صفحة status chart.

	Address	Format	Current Value	New Value
1	I0.1			
2	I0.2			
3	I0.3			
4	I0.4			
5	I0.5			
6	I0.6			

٢– يتم اختيار الصيغة المحددة لتحديد طريقة أظهار حالة المفتاح.

	Address	Format	Current Value	New Value
1	I0.1	Bit		
2	I0.2	Bit		
3	I0.3	Bit		
4	I0.4	Bit		
5	I0.5	Bit		
6	I0.6	Bit		

٣- يتم الضغط على chart status لإظهار حاله المفتاح فى الـ current value.

	Address	Format	Current Value	New Value
1	I0.1	Bit	2#1	
2	I0.2	Bit	2#1	
3	I0.3	Bit	2#0	
4	I0.4	Bit	2#1	
5	I0.5	Bit	2#0	

٤- يتم تطبيق أمر force حلى الدخل "للتشغيل 2#1 أو لإيقاف 2#0".

	Address	Format	Current Value	New Value
1	I0.1	Bit	2#1	2#0
2	I0.2	Bit	2#1	
3	I0.3	Bit	2#0	2#1
4	I0.4	Bit	2#1	
5	I0.5	Bit	2#0	

الاستنتاج.

إذا لم تستجيب وحدة البرمجة إلى أى تعديل من قبل المفتاح فيبقى لنا أن نحدد إذا كانت المشكلة تكمن فى المفتاح ذاته أم فى وحدة البرمجة وبتطبيق الخطوات السابقة يمكن تحديد السبب الرئيسى للعطل ففى حاله أستجابه البرنامج للتعديل الذى تم تطبيقه بواسطة أمر force فهذا يدل على أن المشكلة ليست فى وحدة البرمجة بل فى المفتاح الخارجى أما إن لم تستجيب الوحدة إلى التعديل الذى تم تطبيقه بواسطة أمر force فهذا يدل على أن المشكلة لم تكن فى المفتاح من الأساس بل ان المشكلة تكمن فى وحدة البرمجة أو فى البرنامج ذاته.

يمكن تصميم البرنامج التالى للتأكد من سلامة جميع المداخل بوحدة البرمجة.

يعتمد هذا البرنامج على عدد المدخلات المراد التأكد من سلامتها.

١- إذا كان عدد المدخلات ثمانية فقط فسيكون البرنامج كالتالى:

PROGRAM COMMENTS
Network 1
IB0 ==B 255 —(Q0.1)

الشرح.

للتجربة فقط سيتم غلق الثمانية مفاتيح وإذا كانت جميع المفاتيح سليمة وليست بها أى مشاكل فستكون القيمة الكلية للثمانية مفاتيح هى ٢٥٥ فيغلق مفتاح المقارنة ويعمل الخرج كإشارة تأكيد على سلامة المفاتيح, ولمعرفة سبب اختيار هذا الرقم بالتحديد أنظر صفحة ٨٢ بالجزء الأول من كتاب برمجة التحكم المنطقى PLC.

٢- إذا كان عدد المدخلات ستة عشر فقط فسيكون البرنامج كالتالى:

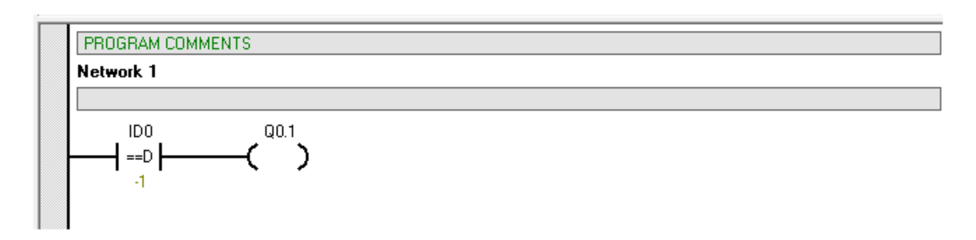
PROGRAM COMMENTS
Network 1
IW0 ==I -1 —(Q0.1)

الشرح.

للتجربة فقط سيتم غلق الستة عشر مفاتيح وإذا كانت جميع المفاتيح سليمة وليست بها أى مشاكل فستكون القيمة الكلية للثمانية مفاتيح هى -١ فيغلق مفتاح المقارنة ويعمل الخرج كإشارة تأكيد على سلامة

المفاتيح, ولمعرفة سبب اختيار هذا الرقم بالتحديد أنظر صفحة ٨٢ بالجزء الأول من كتاب برمجة التحكم المنطقى PLC.

٣- إذا كان عدد المدخلات اثنان وثلاثون فقط فسيكون البرنامج كالتالى:



الشرح.

للتجربة فقط سيتم غلق الاثنان والثلاثون مفتاح وإذا كانت جميع المفاتيح سليمة وليست بها أى مشاكل فستكون القيمة الكلية للاثنين والثلاثون مفاتيح هى -١ فيغلق مفتاح المقارنة ويعمل الخرج كإشارة تأكيد على سلامة المفاتيح, ولمعرفة سبب اختيار هذا الرقم بالتحديد أنظر صفحة ٨٢ بالجزء الأول من كتاب برمجة التحكم المنطقى PLC.

ملاحظة:

- إذا كان عدد المدخلات أكثر من كل ما سبق فسيتم تصميم البرنامج مع استخدام أكثر من مفتاح مقارنة بحيث يتم تغطية جميع المفاتيح.

أعطال المخرجات:

بيان لبعض الأعطال و الأسباب والحلول, للتحقق من المشاكل الخاصة بالمخارج.

إذا كان الحمل يعمل بالخارج يشار إليه بالداخل بأنه يعمل أيضا و الحمل الذى لا يعمل بالخارج يشار إليه بالداخل بأنه لا يعمل أيضا فهذا يدل على أنه لا توجد أى مشكلة بالنسبة للمخارج, إما إذا كان الحمل لا يعمل بالخارج بينما يشير جهاز الـ PLC بأن الحمل يعمل بالداخل فهذا بكل بساطة يشير إلى أنه توجد مشكله بهذا الحمل (لا توجد كهرباء على طرف الريليه – الطرف المشترك على الوحدة غير موصل – تم التوصيل على العزل – خطاء فى توصيل وحدة البرمجة نفسها – مشكله فى نقط الريليه – خطاء فى توصيل دائرة القوة – نقاط القاطع الحرارى مفتوحة) يمكنك النظر إلى طريقة توصيل مختلف أنواع أجهزة الـ PLC فى الجزء الأول من هذا الكتاب.

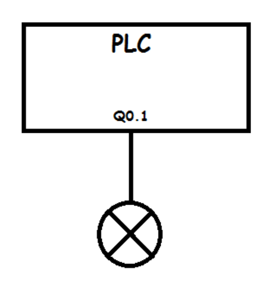
توجد طريقتان لمعرفة سبب العطل أو حل العطل فى حاله أن الخرج يعمل بالفعل ولكن يشار إلى أنه لا يعمل أو أن كان الخرج لا يعمل بالفعل ولكن يشار إلى أنه لا يعمل .

أ –الطريقة الأولى:

يجب تتبع الخطوات حسب الجدولين التاليين حتى التوصل إلى سبب العطل ومن ثم التوصل إلى حل العطل.

الجدول الأول (حينما لا يعمل الخرج بينما تشير وحدة البرمجة إلى أنه يعمل)

فى حالة وجود أى اختلاف بين الحالة الحقيقية للمخرجات وبين الحالة المبينة فى صفحة status chart فهذا يدل على وجود مشكلة,فمثلاً إذا كان الشكل كالتالى:



أن كان الخرج Q0.1 لا يعمل بالخارج كما بالشكل بينما تشير وحدة البرمجة إلى أن الخرج يعمل فهذا يدل بالتأكيد على أن المشكلة هى فى الخرج Q0.1, للتعرف على السبب الأساسى للمشكلة تتبع الخطوات التالية.

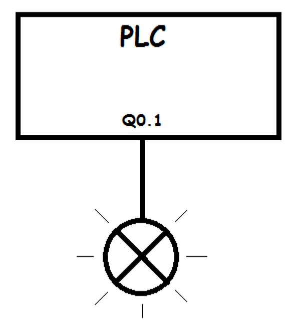
م	السبب	الحل
١	مشكلة بالحمل نفسه	– يجب التأكد من سلامة الحمل. – يجب تغير الحمل.
٢	مشكلة بالسلك نفسه	– يجب التأكد من عدم وجود انقطاع بالسلك. – يجب تغيير السلك.
٣	مشكلة فى ملف الريلية	– يجب التأكد من سلامة ملف الريليه. – يجب تغيير الريليه.
٤	القاطع الحرارى	– التأكد من نقاط القاطع الحرارى over load مغلقة. – الانتظار حتى يبرد القاطع الحرارى وتغلق النقاط تلقائياً.
٥	مشكلة فى التلامس	– يجب إعادة ربط الطرف الموصل على الـ PLC. – يجب إعادة ربط الطرف الموصل على الريليه.
٦	مشكلة فى قيمة الجهد	– يجب التأكد من أن قيمة الجهد لا تقل عن الحد الأدنى بالنسبة للحمل.

٧	مشكلة فى قيمة التيار	– يجب التأكد من أن قيمة التيار لا تزيد عن الحد الأقصى بالنسبة لوحدة البرمجة.
٨	مشكلة فى توصيل وحدة البرمجة	– التأكد من توصيل الطرف المشترك الموجد أو السالب حسب نوع الـ PLC, أنظر صفحة ٣٧ بالجزء الأول من كتاب التحكم المنطقى. – التأكد من توصيل طرف المفتاح بطريقة سليمة على وحدة البرمجة. – التأكد من أنه لم يتم ربط الطرف الموصل على العزل.
٩	مشكلة فى وحدة البرمجة	– التأكد من عدم تطبيق أمر force على الدخل, أنظر صفحة ٥٠ بالجزء الثانى من كتاب التحكم المنطقى. – يجب الضغط على read all forced لمعرفة إذا كان هناك أمر force تم تطبيقه من قبل على أى مفتاح.
١٠	مشكلة فى وحدة البرمجة	– التأكد من عدم وجود عطل فى وحدة المدخلات. – التأكد من أن لمبة diagnostic الخاصة بأعطال وحدة المدخلات لا تضىء (أى لا تشير إلى وجود أعطال).

الجدول الثانى (حينما يعمل الخرج بينما تشير وحدة البرمجة إلى أنه لا يعمل)

فى حالة وجود أى اختلاف بين الحالة الحقيقية للمخرجات وبين الحالة المبينة فى صفحة status chart فهذا يدل على وجود مشكلة,فمثلاً إذا كان الشكل كالتالى:

أن كان الخرج Q0.1 يعمل بالخارج كما بالشكل بينما تشير وحدة البرمجة إلى أن الخرج لا يعمل فهذا يدل بالتأكيد على أن المشكلة هى فى الخرج Q0.1, للتعرف على السبب الأساسى للمشكلة تتبع الخطوات التالية.



م	السبب	الحل
١	مشكلة بالحمل نفسه	يجب التأكد من سلامة الحمل.
٢	مشكلة فى ملف الريلية	يجب التأكد من سلامة ملف الريليه.
٣	مشكلة فى نقط التلامس	يجب التأكد من التوصيل على النقطة المفتوحة للريليه وليس المغلقة.
٤	مشكلة فى توصيل وحدة البرمجة	التأكد من توصيل الطرف المشترك الموجد أو السالب حسب نوع ال PLC, أنظر صفحة ٣٧ بالجزء الأول من كتاب التحكم المنطقى.
٥	مشكلة فى وحدة البرمجة	التأكد من عدم تطبيق أمر force على الدخل, أنظر صفحة ٥٠ بالجزء الثانى من كتاب التحكم المنطقى.
٦	مشكلة فى وحدة البرمجة	التأكد من عدم وجود عطل فى وحدة المدخلات.

ب الطريقة الثانية:

يجب تتبع الخطوات التالية حتى التوصل إلى معرفة إذا كان سبب العطل هو بسبب مشكلة خارج أم داخل وحدة البرمجة.

١– يكتب عنوان الخرج المشكوك فيه داخل الـ address في صفحة status chart.

	Address	Format	Current Value	New Value
1	Q0.1			
2	Q0.2			
3	Q0.3			
4	Q0.4			
5	Q0.5			
6	Q0.6			

٢– يتم اختيار الصيغة المحددة لتحديد طريقة أظهار حالة المفتاح.

	Address	Format	Current Value	New Value
1	Q0.1	Bit		
2	Q0.2	Bit		
3	Q0.3	Bit		
4	Q0.4	Bit		
5	Q0.5	Bit		
6	Q0.6	Bit		

٣– يتم الضغط على chart status لإظهار حاله المفتاح في الـ current value.

	Address	Format	Current Value	New Value
1	Q0.1	Bit	2#0	
2	Q0.2	Bit	2#1	
3	Q0.3	Bit	2#1	
4	Q0.4	Bit	2#0	
5	Q0.5	Bit	2#1	

٤- يتم تطبيق أمر force على الخرج "للتشغيل 2#1 أو لإيقاف 2#0".

	Address	Format	Current Value	New Value
1	Q0.1	Bit	2#0	2#1
2	Q0.2	Bit	2#1	
3	Q0.3	Bit	2#1	2#0
4	Q0.4	Bit	2#0	
5	Q0.5	Bit	2#1	

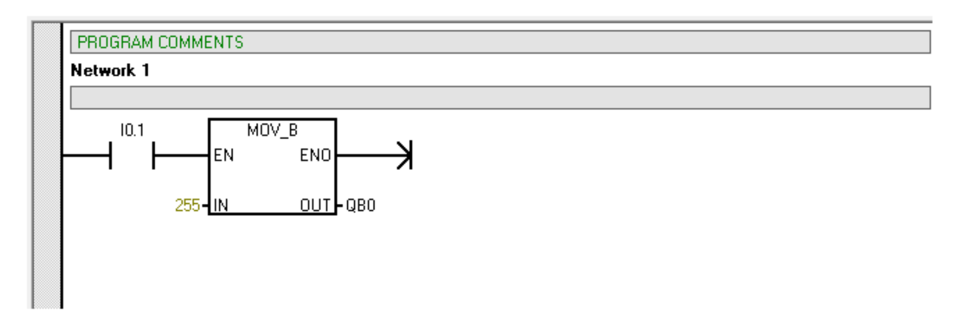
الاستنتاج.

إذا لم يستجيب الخرج إلى أى تعديل من قبل وحدة البرمجة فيبقى لنا أن نحدد إذا كانت المشكلة تكمن فى الخرج ذاته أم فى وحدة البرمجة وبتطبيق الخطوات السابقة يمكن تحديد السبب الرئيسى للعطل ففى حاله أستجاب البرنامج للتعديل الذى تم تطبيقه بواسطة أمر force فهذا يدل على أن المشكلة ليست فى وحدة البرمجة بل فى الحمل أما إن لم تستجيب الوحدة إلى التعديل الذى تم تطبيقه بواسطة أمر force فهذا يدل على أن المشكلة لم تكن فى الحمل من الأساس بل ان المشكلة تكمن فى وحدة البرمجة أو فى البرنامج ذاته.

يمكن تنفيذ البرنامج التالى التأكد من سلامة جميع المخارج بوحدة البرمجة.

يعتمد هذا البرنامج على عدد المخارج المراد التأكد من سلامتها.

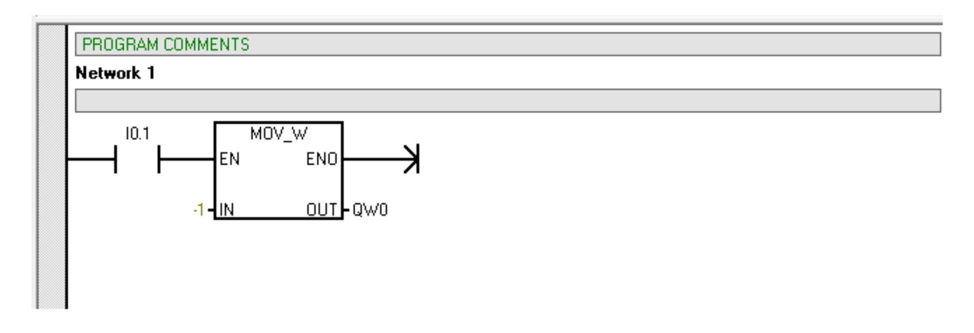
١- إذا كان عدد المخارج ثمانية فقط فسيكون البرنامج كالتالى:



الشرح.

للتجربة فقط سيتم غلق المفتاح I0.1 لكى يقوم بنقل قيمة ٢٥٥ بواسطة النقلات إلى الثمانية مخارج فإذا أضاءت جميع الثمانية مخارج فهذا دليل على أن جميعها سليم وليست هناك أى مشاكل, ولمعرفة سبب اختيار هذا الرقم بالتحديد أنظر صفحة ٨٢ بالجزء الأول من كتاب برمجة التحكم المنطقى PLC.

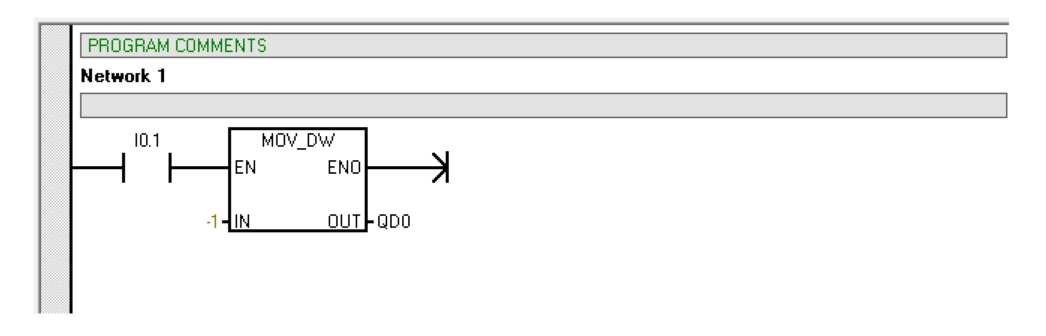
٢- إذا كان عدد المخارج ستة عشر فقط فسيكون البرنامج كالتالى:



الشرح.

للتجربة فقط سيتم غلق المفتاح I0.1 لكى يقوم بنقل قيمة -١ بواسطة النقلات إلى الستة عشرة مخرج فإذا أضاءت جميع الستة عشرة مخرج فهذا دليل على أن جميعها سليم وليست هناك أى مشاكل, ولمعرفة سبب اختيار هذا الرقم بالتحديد أنظر صفحة ٨٢ بالجزء الأول من كتاب برمجة التحكم المنطقى PLC.

٣- إذا كان عدد المخارج أثنين وثلاثون فقط فسيكون البرنامج كالتالى:



الشرح.

للتجربة فقط سيتم غلق المفتاح I0.1 لكى يقوم بنقل قيمة -١ بواسطة النقلات إلى الاثنين والثلاثون مخرج فإذا أضاءت جميع الاثنين والثلاثون مخرج فهذا دليل على أن جميعها سليم وليست هناك أى مشاكل, ولمعرفة سبب اختيار هذا الرقم بالتحديد أنظر صفحة ٨٢ بالجزء الأول من كتاب برمجة التحكم المنطقى PLC.

ملاحظة:

- إذا كان عدد المخارج أكثر من كل ما سبق فسيتم تصميم البرنامج مع استخدام عدد أكثر من النقلات بحيث يتم تغطية جميع المخارج.
- فى حالة تطبيق أى من الدوائر السابقة يجب التأكد بحزم من فصل جميع دوائر القوة حتى لا تعمل المحركات بالمصنع أثناء التجربة لأن هذا قد يؤدى إلى عواقب وخيمة.

الباب الثانى

الرسائل التحذيرية

- مقدمة عن الرسائل التحذيريـــــــــــــــة.
- الخطوات الأساسية لتنفيذ الرسائل التحذيريــــة.
- الرسائل التحذيرية لمحرك يعمل فى إتجاه واحد.
- الرسائل التحذيرية لمحرك يعمل فى إتجاهيـــن.
- الرسائل التحذيريـــــــة للمعادلات الرياضية.
- أمثله مختلفة باستخدام الرسائل التحذيريــــــة.

رسائل تحذيرية:

من الحلول الممكنة لسهولة التعرف على الأعطال المتواجدة بالبرنامج هى أن نجعل وحدة البرمجة تقوم بإظهار رسائل تدل على الأعطال علماً بأن وحدة البرمجة تظهر رسائل خاصة بأخطاء معينة وليس بأى نوع من الأعطال لأنة ليس من الممكن لوحدة البرمجة أن تكتشف وحدها أن المحرك الاتجاهين لا يمكن أن يعمل فى الاتجاهين معاً لأن فى الأساس هو لا يعرف أساساً ما معنى محرك اتجاهين ولكن يمكننا تصميم برنامج بحيث فى حاله تشغيل الاتجاهين معاً يقوم الـ PLC بإظهار رسالة تدل على حدوث "قفله" وهذا ما سيتم توضيحه فى المثال التالى

التمرين الأول:

محرك يعمل فى اتجاهين بحيث أنه فى حالة أن حاول أحد تشغيل الاتجاهين معاً بالضغط على مفاتيح التشغيل معاً تظهر رسائل تحذيرية تشير إلى أن لا يمكن للمحرك أن يعمل فى الاتجاهين فى نفس الوقت أما أن قام أحد بتشغيل الخرجين معاً بالأجبار force فتظهر رسالة تدل على أنه حدثت قفله.

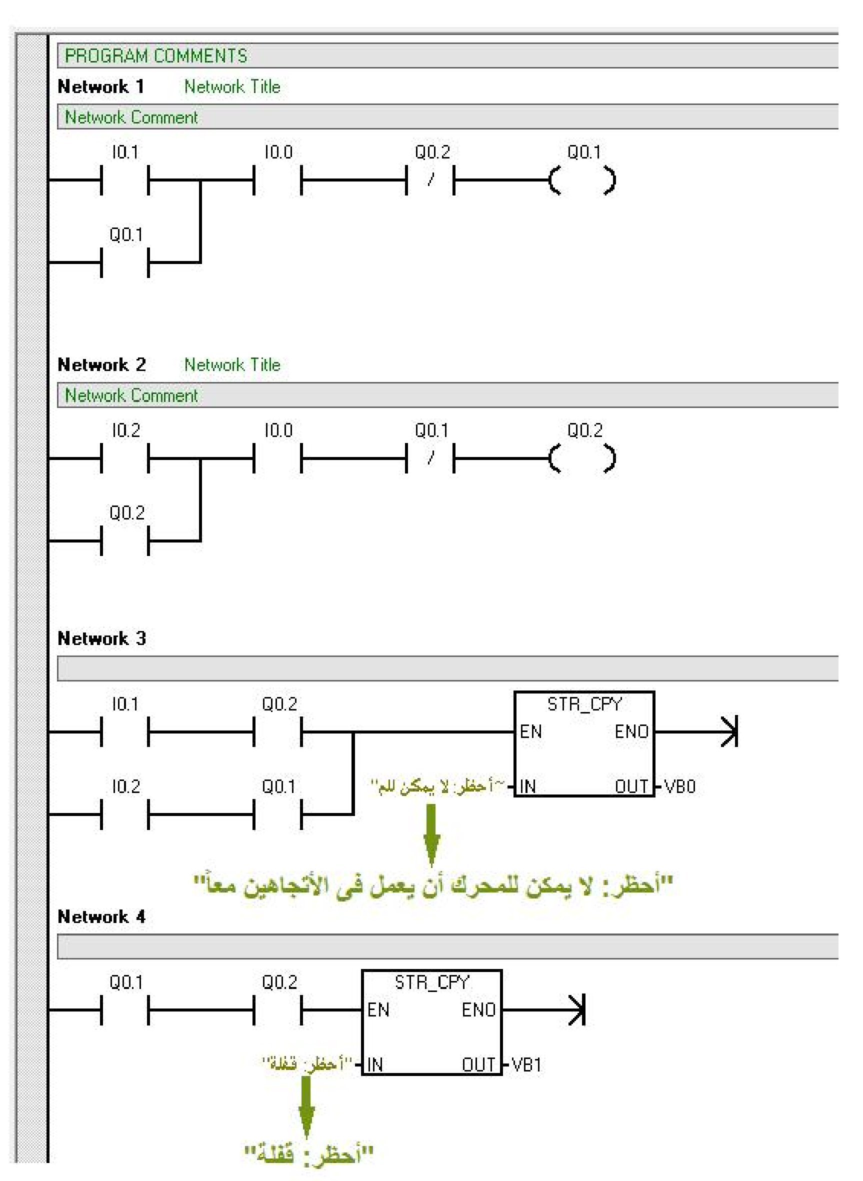
الخطوات:

- يتم أولاً تنفيذ الجزء الخاص بالمحرك عكس حركة "الفرع الأول والثانى".
- يتم بعد ذالك تنفيذ الجزء الخاص بالرسائل التحذيرية "الفرع الثالث والرابع".

ملاحظة:

لتجنب حدوث القفله الكهربية من الأساس يرجى وضع حمايات على ريليهات الحماية خارج وحدة البرمجة ولمعرفة المزيد عن ريليهات الحماية أنظر صفحة ٣٢ بالجزء الأول من كتاب برمجة التحكم المنطقى PLC.

البرنامج:


PROGRAM COMMENTS
Network 1
Network Title
Network Comment
I0.1
I0.0
Q0.2
Q0.1
Q0.1
Network 2
Network Title
Network Comment
I0.2
I0.0
Q0.1
Q0.2
Q0.2
Network 3
I0.1
Q0.2
STR_CPY
EN
ENO
I0.2
Q0.1
IN
OUT
VB0
"أحظر: لا يمكن للمحرك أن يعمل فى الأتجاهين معاً"
Network 4
Q0.1
Q0.2
STR_CPY
EN
ENO
IN
OUT
VB1
"أحظر: قفلة"

الشرح.

الفرع الأول:

تم رسم الجزء الخاص بعمل المحرك فى اتجاه محدد.

الفرع الثانى:

تم رسم الجزء الخاص بعمل المحرك فى الاتجاه الأخر.

الفرع الثالث:

تم تنفيذ الجزء الخاص بالرسالة التحذيرية فى حالة أن حاول أحد تشغيل المحرك فى الاتجاهين معاً, إذا قام أحد بالضغط على مفتاح تشغيل أى اتجاه أثناء عمل الاتجاه الأخر تظهر الرسالة التالية على الذاكرة VB0: " أحظر: لا يمكن للمحرك أن يعمل فى الاتجاهين معاً ".

الفرع الرابع:

تم تنفيذ الجزء الخاص بالرسالة التحذيرية الثانية أى أنه فى حالة أن قام أحد بتشغيل المحرك فى الاتجاهين معاً مثلاً باستخدام أمر force, تظهر الرسالة التالية على الذاكرة VB1: " أحظر: قفلة ".

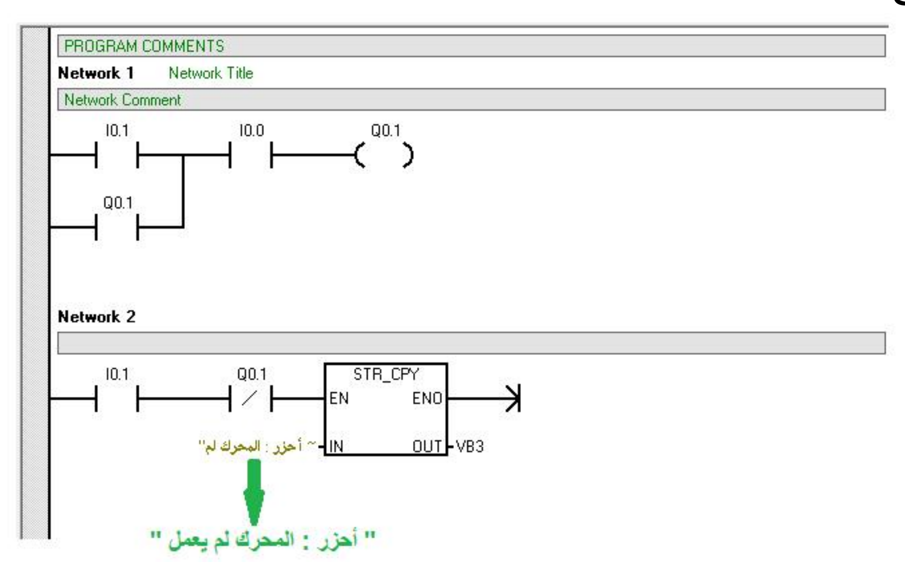
التمرين الثانى:

محرك يعمل بطريقة طبيعية جداً أى أنه عند الضغط على مفتاح التشغيل ولكن إذ لم يعمل المحرك عند الضغط على مفتاح التشغيل فهذا يدل على وجود خطاء ما فيجب أن تظهر رسالة لكى تشير بأن المحرك الفلانى لم يعمل.

الخطوات:

- يتم أولاً تنفيذ الجزء الخاص بالمحرك "الفرع الأول".
- يتم بعد ذالك تنفيذ الجزء الخاص بالرسالة التحذيرية "الفرع الثانى".

البرنامج:



ملاحظة:

لمزيد من الواقعية يفضل أن يكون هناك مصدر خارجى لمعرفة إذا كان المحرك يعمل بالفعل أم لا لأنه من الممكن جداً أن يعمل الخرج على وحدة البرمجة ولكن لظروف خارجية معينة لا يعمل المحرك فيفضل استخدام مفاتيح الطرد المركزية التى تغلق تلقائياً عندما يعمل المحرك وباستخدام هذا النوع من المفاتيح يمكن التأكد من عمل المحرك بطريقة سليمة.

الشرح.

محرك يعمل بطريقة طبيعية جداً أى عند الضغط على مفتاح التشغيل ولكن إذ لم يعمل المحرك عند الضغط على مفتاح التشغيل فهذا يدل على وجود خطاء ما فيجب أن تظهر رسالة لكى تشير بأن المحرك الفلانى لم يعمل.

الفرع الأول:

تم رسم الجزء الخاص بعمل المحرك.

الفرع الثانى:

تم تنفيذ الجزء الخاص بالرسالة التحذيرية فى حالة أن قام أحد بالضغط على مفتاح البدء لتشغيل المحرك ولكن لظروف معينة لا يعمل المحرك منها مثلاً وجود مشكلة بمفتاح الإيقاف فتظهر الرسالة التالية على الذاكرة VB3: " أحظر: المحرك لم يعمل ".

التمرين الثالث:

محرك يعمل بطريقة طبيعية جداً أى عند الضعط على مفتاح التشغيل كما بالمثال السابق ولكن إذ لم يعمل المحرك عند الضغط على مفتاح التشغيل فهذا يدل على وجود خطاء ما فيجب أن تظهر رسالة لكى تشير بأن كان سبب المشكلة داخل أم خارج وحدة البرمجة.

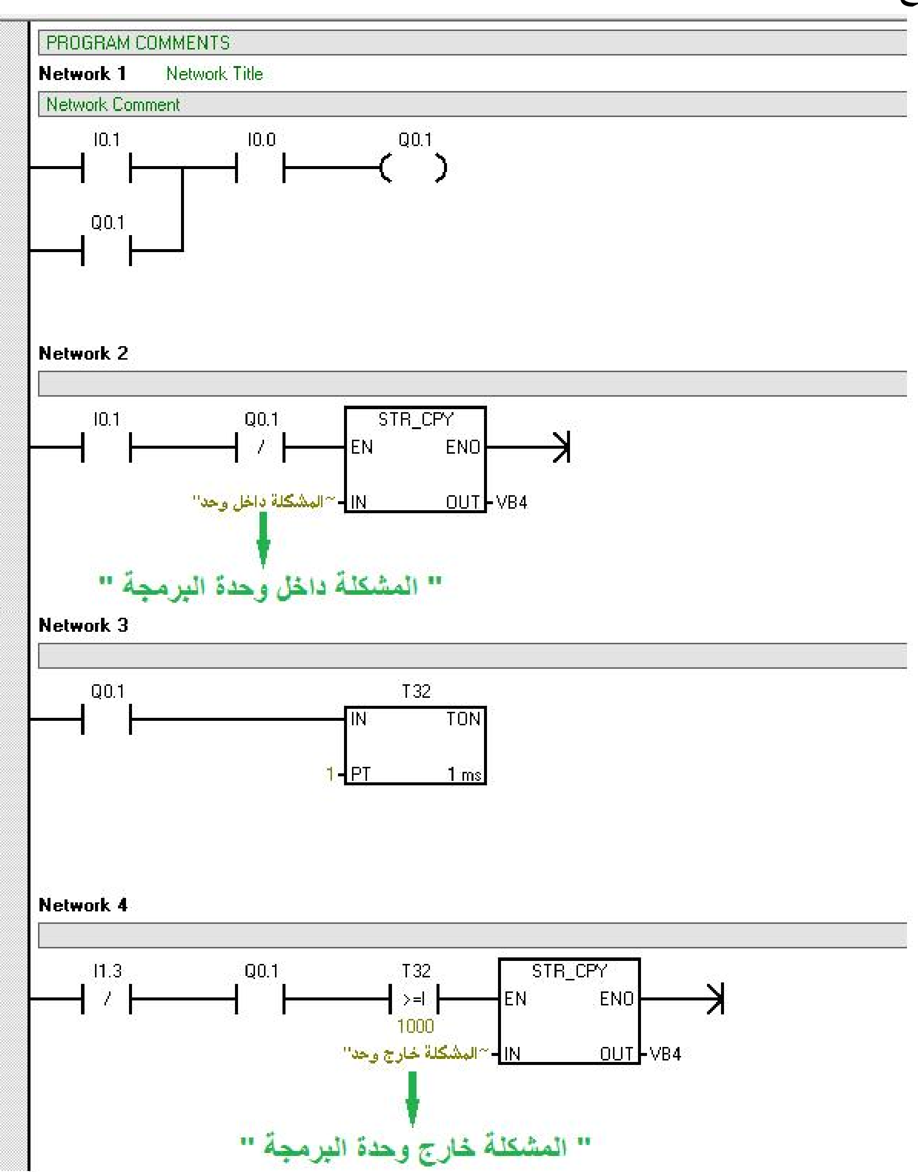
الخطوات:

- يتم أولاً تنفيذ الجزء الخاص بالمحرك "الفرع الأول".
- يتم بعد ذالك تنفيذ الجزء الخاص بالرسائل التحذيرية "الفرع الثانى والرابع".

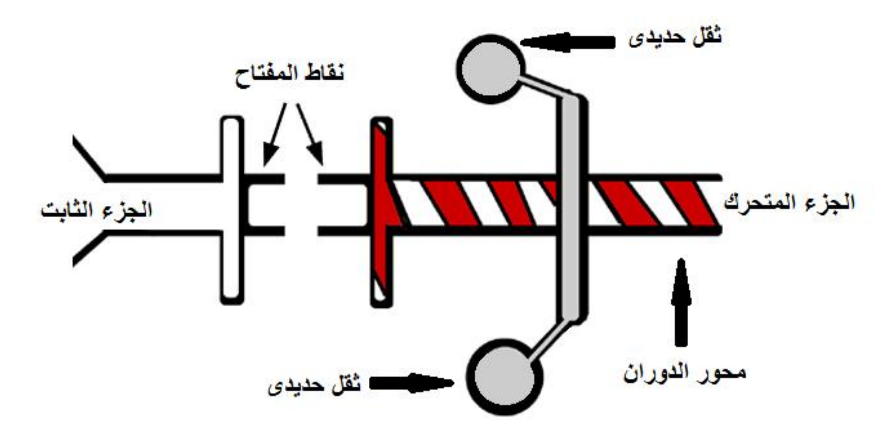
ملاحظة:

يحتاج هذا التمرين إلى استخدام مفتاح يثبت على محور المحرك ويغلق تلقائياً بفعل قوة الطرد المركزية Centrifugal switch عندما يعمل المحرك كما هو ممثل بالتمرين التالى بواسطة I1.3.

البرنامج:


PROGRAM COMMENTS
Network 1
Network Title
Network Comment
I0.1
I0.0
Q0.1
Q0.1
Network 2
I0.1
Q0.1
STR_CPY
EN
ENO
"المشكلة داخل وحد"~
IN
OUT
VB4
" المشكلة داخل وحدة البرمجة "
Network 3
Q0.1
T32
IN
TON
1
PT
1 ms
Network 4
I1.3
Q0.1
T32
>=I
1000
STR_CPY
EN
ENO
"المشكلة خارج وحد"~
IN
OUT
VB4
" المشكلة خارج وحدة البرمجة "

رسم **توضيحى**:



الشرح.

الفرع الأول:

تم رسم الجزء الخاص بعمل المحرك.

الفرع الثانى:

تم تنفيذ الجزء الخاص بالرسالة التحذيرية فى حالة أن كان المحرك لم يعمل بسبب مشكلة داخل وحدة البرمجة فتظهر الرسالة التالية على الذاكرة VB4: " المشكلة داخل وحدة البرمجة ".

الفرع الثالث:

يتم احتساب زمن معين بداية من تشغيل المحرك وحتى أظهار الرسالة التحذيرية التالية.

الفرع الرابع:

تم تنفيذ الجزء الخاص بالرسالة التحذيرية فى حالة أن كان المحرك لم يعمل بسبب مشكلة خارج وحدة البرمجة فتظهر الرسالة التالية على نفس الذاكرة VB4: " المشكلة خارج وحدة البرمجة ".

التمرين الرابع:

قم بتنفيذ معادلة تقوم بتحويل درجة الحرارة من الـ Fahrenheit إلى الـ Celsius بشرط أن تقوم وحدة البرمجة بإظهار رسائل تحذيرية توضح الحالات:

١- درجة الحرارة أكثر من ١٠٠ درجة مئوية.

٢- درجة الحرارة أقل من الصفر درجة مئوية (بالسالب).

٣- درجة الحرارة صفر درجة مئوية.

الخطوات:

- يتم أولاً تحديد المتغيرات داخل "صفحة الرموز".
- يتم ثانياً تنفيذ الجزء الخاص بالمعادلة "الفرع الأول والثانى".
- يتم بعد ذالك تنفيذ الجزء الخاص بالرسائل التحذيرية "الفرع الثالث والرابع والخامس".

ملاحظة:

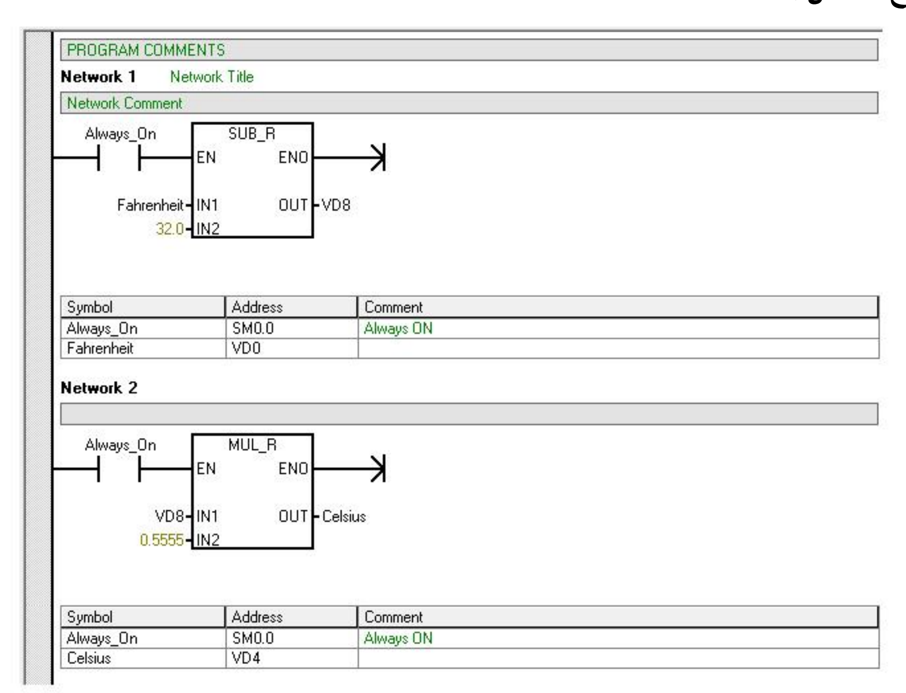
- يحتاج هذا التمرين إلى استخدام المفاتيح المرتبطة بالريليهات الخاصة الـ special markers, لمعرفة المزيد عن هذا النوع من الريليهات أنظر صفحة ١٣٥بالجزء الثانى من كتاب برمجة التحكم المنطقى PLC.
- لا يمكن لوحدة البرمجة أن تفهم ما المقصود بالـ Fahrenheit و بالـ Celsius لذالك يجب تحديد عنوان لكلن منهم باستخدام "صفحة الرموز", لمزيد من التفاصيل أنظر صفحة ٩٥ بالجزء الثانى من كتاب برمجة التحكم المنطقى PLC.
- القانون الرياضى للتحويل من الـ Fahrenheit إلى الـ Celsius هو التالى:

$$C = (F-32) * (5/9)$$

صفحة الرموز:

		Symbol	Address	Comment
1		Fahrenheit	VD0	
2		Celsius	VD4	

البرنامج الخاص بالمعادلة:



Symbol	Address	Comment
Always_On	SM0.0	Always ON
Fahrenheit	VD0	

Symbol	Address	Comment
Always_On	SM0.0	Always ON
Celsius	VD4	

الشرح.

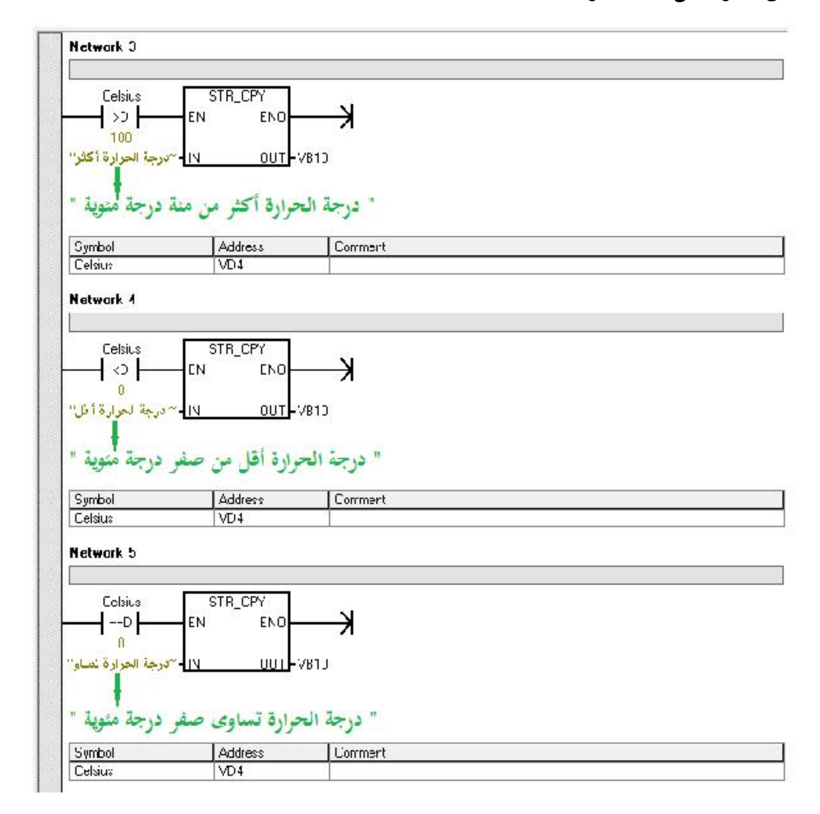
الفرع الأول:

تم رسم الجزء الخاص بعمل المحرك.

الفرع الثانى:

تم تنفيذ الجزء الخاص بالرسالة التحذيرية فى حالة أن كان المحرك لم يعمل بسبب مشكلة داخل وحدة البرمجة فتظهر الرسالة التالية على الذاكرة VB4: " المشكلة داخل وحدة البرمجة ".

البرنامج الخاص بالرسائل التحذيرية:



الشرح.

الفرع الثالث:

يتم تنفيذ الجزء الخاص بالرسالة التحذيرية فى حالة أن كانت درجة الحرارة أكثر من مئة درجة مئوية فتظهر الرسالة التالية على الذاكرة VB10: " درجة الحرارة أكثر من مئة درجة مئوية ".

الفرع الرابع:

يتم تنفيذ الجزء الخاص بالرسالة التحذيرية فى حالة أن كانت درجة الحرارة أقل من صفر درجة مئوية فتظهر الرسالة التالية على الذاكرة VB10: " درجة الحرارة أقل من صفر درجة مئوية ".

الفرع الخامس:

يتم تنفيذ الجزء الخاص بالرسالة التحذيرية فى حالة أن كانت درجة الحرارة تساوى صفر درجة مئوية فتظهر الرسالة التالية على الذاكرة VB10: " درجة الحرارة تساوى صفر درجة مئوية ".

التمرين الخامس:

قم بتنفيذ المعادلة التالية:

$$Y = 3X^2 - 0.5X + 10$$

بحيث أنه يمكن تغيير قيمة الـ X بواسطة عداد خارجى وتقوم وحدة البرمجة بحساب قيمة Y ومن ثم يقوم بعرض ناتج المعادلة Y على شاشة الـ seven segment display وبشرط أن تقوم وحدة البرمجة بإظهار رسائل تحذيرية فى الحالات التالية:

١- إذا كانت القيمة بالسالب.
٢- إذا كانت القيمة تساوى صفر.
٣- إذا تمت القسمة على صفر.
٤- إذا وجدت مشكلة بحجم الذاكرة.
٥- إذا تم إدخال قيمة BCD خطاء.

الخطوات:

- يتم أولاً تنفيذ الجزء الخاص بأدخال القيمة إلى وحدة البرمجة.
- يتم ثانياً تنفيذ الجزء الخاص بالمعادلة.
- يتم ثالثاً تنفيذ الجزء الخاص بأخراج القيمة على شاشة عرض SSD.
- يتم فى النهاية تنفيذ الجزء الخاص بالرسائل التحذيرية كما ذكرت بالأعلى.

ملاحظة:

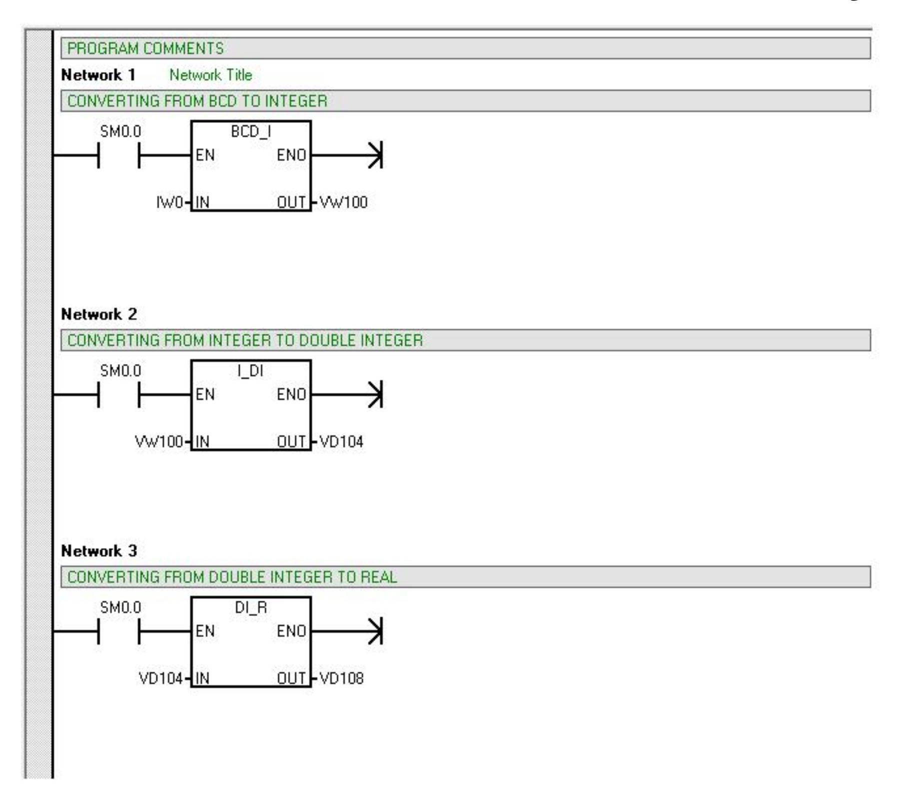
- يحتاج هذا التمرين إلى استخدام المفاتيح المرتبطة بالريليهات الخاصة الـ special markers, لمعرفة المزيد عن هذا النوع من الريليهات أنظر صفحة ١٣٥بالجزء الثانى من كتاب برمجة التحكم المنطقى PLC.
- أحرص أن تقوم أولاً بتنفيذ العمليات ذات الأولوية مثل الضرب والقسمة ومن ثم العمليات الأقل أولوية مثل الجمع والطرح.
- يفضل تقسيم المعادلة كما فى الجدول التالى ليصبح من السهل بعد ذالك تنفيذ التمرين فعلياً.
- لا يمكن لوحدة البرمجة أن تفهم ما المقصود بالـ X و بالـ Y لذالك يجب تحديد عنوان لكل منهم باستخدام "صفحة الرموز" كما بالمثال السابق, لمزيد من التفاصيل أنظر صفحة ٩٥ بالجزء الثانى من كتاب برمجة التحكم المنطقى PLC.

الجدول.

م	المعادلة	الخطوات	الذاكرة	النواتج
١	X^2	X . X	VD0 . VD0	VD8
٢	$3X^2$	$3 . X^2$	3 . VD8	VD12
٣	0.5X	0,5 . X	0,5 . VD0	VD16
٤	$3X^2 - 0.5X$	$(3 . X^2) – (0,5 . X)$	VD12 – VD16	VD20
٥	$3X^2 - 0.5X + 10$	(3 . X2 - 0,5 . X + 10)	VD20 + 10	VD4

البرنامج:

الجزء الخاص بإدخال القيمة إلى وحدة البرمجة.



الشرح.

الفرع الأول:

يتم استقبال القيمة الـ BCD من العداد الخارجى وتحويلها إلى قيمة Integer.

الفرع الثانى:

يتم تغيير حجم الذاكرة من الـ Word إلى الـ Double word.

الفرع الثالث:

يتم تحويل القيمة من رقم صحيح إلى رقم عشرى عن طريق إضافة صفر بعد العلامة.

الجزء الخاص بالمعادلة الرياضية.

Network 4

THE EQUATION

SM0.0 — MUL_R: EN / ENO; VD0 – IN1; VD0 – IN2; OUT – VD8

Network 5

THE EQUATION

SM0.0 — MUL_R: EN / ENO; 3.0 – IN1; VD8 – IN2; OUT – VD12

Network 6

THE EQUATION

SM0.0 — MUL_R: EN / ENO; 0.5 – IN1; VD0 – IN2; OUT – VD16

Network 7

THE EQUATION

SM0.0 — SUB_R: EN / ENO; VD12 – IN1; VD16 – IN2; OUT – VD20

Network 8

THE EQUATION

SM0.0 — ADD_R: EN / ENO; VD20 – IN1; 10.0 – IN2; OUT – VD4

الشرح.

الفرع الرابع:

يتم تنفيذ الجزء الخاص بـ X^2

الفرع الخامس:

يتم تنفيذ الجزء الخاص بـ $3X^2$

الفرع السادس:

يتم تنفيذ الجزء الخاص بـ $0.5X$

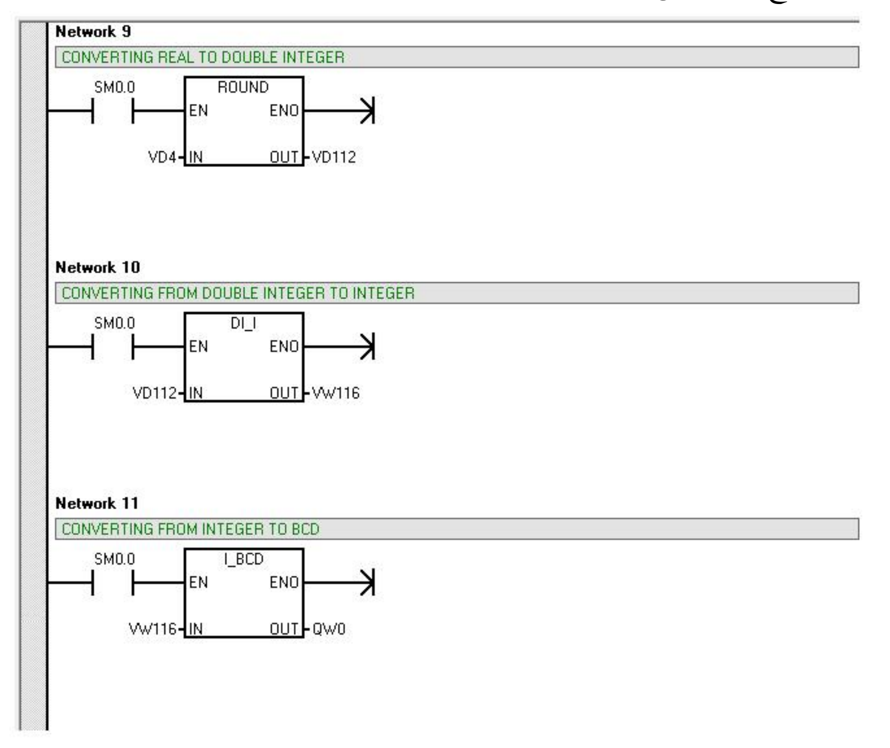
الفرع السابع:

يتم تنفيذ الجزء الخاص بـ $3X^2 - 0.5X$

الفرع الثامن:

يتم تنفيذ الجزء الخاص بـ $3X^2 - 0.5X + 10$

الجزء الخاص بإخراج القيمة على شاشة عرض.



الشرح.

الفرع التاسع:

يتم تحويل الرقم العشرى إلى رقم صحيح عن طريق التقريب إلى أقرب رقم صحيح.

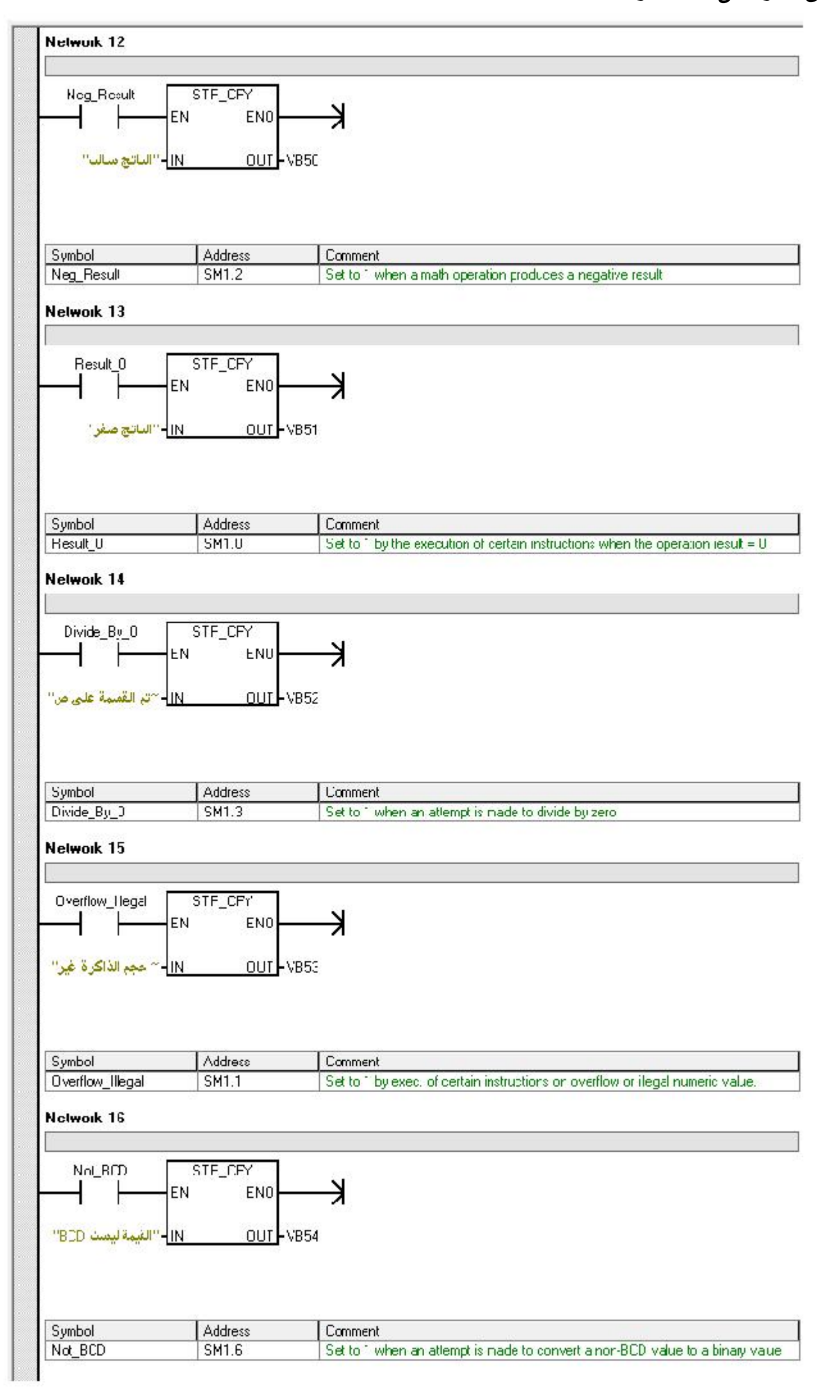
الفرع العاشر:

يتم تغيير حجم الذاكرة من الـ Double word إلى الـ word.

الفرع الحادى عشر:

يتم تحويل القيمة الـ Integer إلى قيمة BCD ومن ثم إظهارها على الشاشة.

الجزء الخاص بالرسائل التحذيرية.



الشرح.

الفرع الثانى عشر:

يتم تنفيذ الجزء الخاص بالرسالة التحذيرية فى حالة أن كان ناتج المعادلة سالب فتظهر الرسالة التالية على الذاكرة VB50: " الناتج سالب ".

الفرع الثالث عشر:

يتم تنفيذ الجزء الخاص بالرسالة التحذيرية فى حالة أن كان ناتج المعادلة صفر فتظهر الرسالة التالية على الذاكرة VB51: " الناتج صفر ".

الفرع الرابع عشر:

يتم تنفيذ الجزء الخاص بالرسالة التحذيرية فى حالة أن تمت القسمة على صفر فتظهر الرسالة التالية على الذاكرة VB52: " تمت القسمة على صفر ".

الفرع الخامس عشر:

يتم تنفيذ الجزء الخاص بالرسالة التحذيرية فى حالة أن وجدت مشكلة فى الذاكرة فتظهر الرسالة التالية على الذاكرة VB53: " حجم الذاكرة غير كافى ".

الفرع السادس عشر:

يتم تنفيذ الجزء الخاص بالرسالة التحذيرية فى حالة أن كانت قيمة العداد الخارجى غير صحيحة فتظهر الرسالة التالية على الذاكرة VB54: " قيمة العداد الخارجى ليست BCD ".

التمرين السادس:

قم بتنفيذ تمرين يحتوى على عداد بحيث يتم تغيير القيمة المحددة بواسطة عداد خارجى بشرط أن تقوم وحدة البرمجة بإظهار رسائل تحذيرية توضح الحالات التالية:

١- إذا كانت القيمة أكثر من ٣٢٧٦٧ وهو أقصى رقم يمكن استخدامه مع العداد .
٢- إذا كانت القيمة تساوى صفر.
٣- إذا كانت القيمة المدخلة من قبل العداد غير صحيحة ليست BCD.

الخطوات:

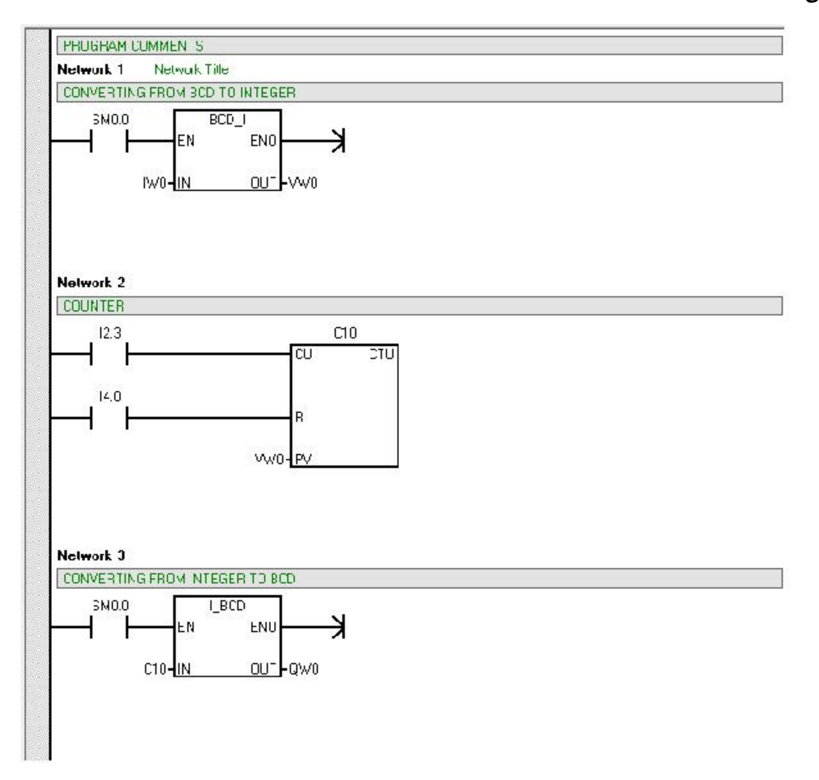
- يتم أولاً تنفيذ الجزء الخاص بالمعادلة.
- يتم بعد ذالك تنفيذ الجزء الخاص بالرسائل التحذيرية.

ملاحظة:

- يحتاج هذا التمرين إلى استخدام المفاتيح المرتبطة بالريليهات الخاصة الـ special markers, لمعرفة المزيد عن هذا النوع من الريليهات أنظر صفحة ١٣٥بالجزء الثانى من كتاب برمجة التحكم المنطقى PLC.
- تعتبر قيمة العداد الخارجى BCD لذالك إذا كانت القيمة المدخلة من نوع أخر تقوم وحدة البرمجة بأصدار رسالة تحذيرية.

البرنامج:

الجزء الخاص بالعداد.



الشرح.

الفرع الأول:

يتم استقبال القيمة الـ BCD من العداد الخارجى وتحويلها إلى قيمة Integer.

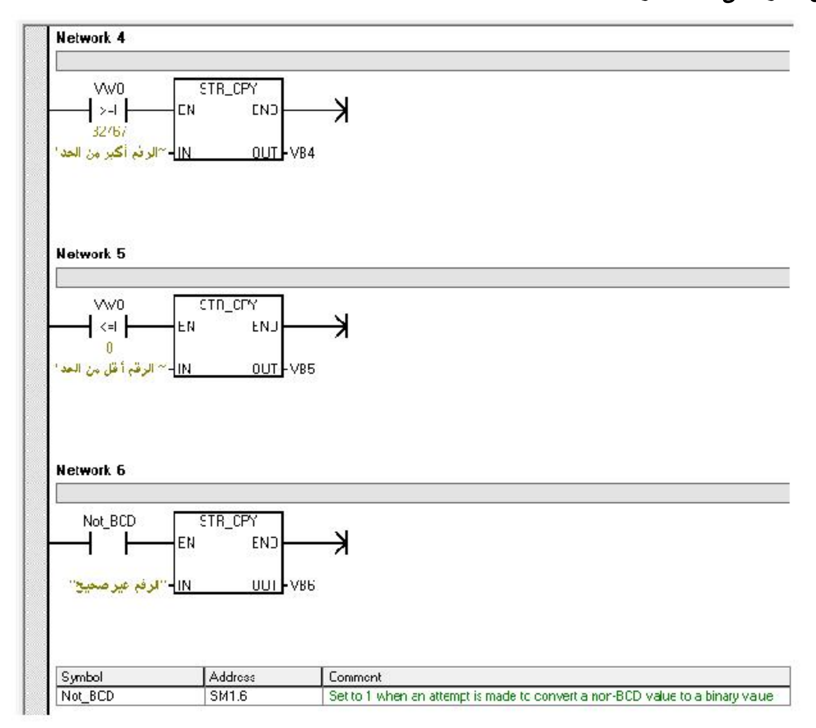
الفرع الثانى:

يقوم العداد C10 باحتساب تلك القيمة كقيمة مسبقة للاستخدام فيما بعد.

الفرع الثالث:

يتم تحويل القيمة الـ Integer إلى قيمة BCD ومن ثم إظهارها على الشاشة SSD.

الجزء الخاص بالرسائل التحذيرية.



الشرح.

الفرع الرابع:

فى حالة أن كانت القيمة أكبر من ٣٢٧٦٧ فتظهر الرسالة التالية على الذاكرة VB5: "الرقم أكبر من الحد الأقصى".

الفرع الخامس:

فى حالة أن كانت القيمة تساوى صفر فتظهر الرسالة التالية على الذاكرة VB5: "الرقم أقل من الحد الأدنى".

الفرع السادس:

كما فى المثال السابق.

الباب الثالث

أنواع الأعطال

- شرح أنواع الأعطـــــــــــــــــــــــــال.
- النـــــــــــــــــوع الأول **Compile errors**.
- النـــــــــــــــــوع الثانى **Non Fatal errors**.
- النـــــــــــــــــوع الثالث **Fatal errors**.
- صفحـــــــــات المساعـــــــــــــدة **Help**.
- صفحات الريليهات الخاصة **Special Markers**.
- تحديد الأفرع التى تحتوى على إخطـــــــــــــاء.
- إصـــــــــــــــلاح مشاكل الـ **hardware**.
- الصفح المستخدمة لكشف الأعطـــــــــــال.
- بعض أنواع الأعطـــــــــــــــــال المتدولة.
- أمثله عملية قبل و بعد تصحيح العطـــــــــــل.

أنواع الأعطال وإصلاحها داخل وحدة البرمجة:

بسبب أن وحدة البرمجة تركب وتبرمج من قبل الناس فقد توجد أخطاء ناتجة عن قلة الانتباه أو قلة الخبرة. فلذالك تذود أجهزة الـ PLC بمميزات كثيرة لإيجاد هذه الأخطاء . يمكن تقسيم هذه الأخطاء إلى ثلاث أنواع:

النوع الأول:

وهى من أكثر الأنواع التى يتم التعرض لها داخل وحدة البرمجة وتسمى بالـ compile errors حيث قد يجتاز عدد الأعطال من هذا النوع المئات ولكن لا يبدو الموضوع صعب أو معقد لأن وحدة البرمجة تقوم بمساعدتك فى اكتشاف وحل العطل قبل حتى أن يتم نقل البرنامج إلى وحدة البرمجة وهي لا تكتشف من قبل وحدة البرمجة بشكل آلي ولكن يمكن اكتشافها باستخدام تعليمات اكتشاف الأخطاء فتعطى وحدات البرمجة رسائل تحذير حيث يقوم البرنامج بمراجعة التمرين المرسوم للتأكد من عدم وجود إى أعطال, إما فى حالة وجود أعطال بسبب أن المبرمج كتب برنامجاً فيه خطأ واضح فسيساعدك البرنامج على اكتشاف الأعطال كما سنوضح فى المثال التالى.

المساعدة المقدمة من قبل وحدة البرمجة للمساعدة فى اكتشاف وحل النوع الأول من الأعطال تتمثل فى الثلاث نقاط التالية:

- فأن البرنامج يقوم تلقائياً بتحديد نوع, أسم, رقم, سبب, مكان العطل بل ويقترح عليك الحل أيضاً.
- يتم تغير لون الاسم الخطاء من اللون الأسود إلى اللون الأحمر لكى يشير بأنه غير صحيح.
- بالضغط على الرسالة التى تشير إلى مكان العطل يتوجه البرنامج تلقائياً إلى الرمز أو الاسم الخطاء ويقوم بعمل مربع حول الخطاء.

مثال على النوع الأول لمحرك يعمل من مكان ويقف من مكان

أولاً: يتم رسم البرنامج باستخدام أى من لغات البرمجة.

PROGRAM COMMENTS
Network 1 Network Title
Network Comment
I0.1 I0.0 q0.9
Q0.0

ملاحظة:

في حالة كتابة أى مسميات بطريقة خطاء سوف تظهر تلقائياً باللون الأحمر ولكن لا تلاحظ لأن الكتاب ليس مطبوع بالألوان.

ثانياً: يتم الضغط على أمر compile لمعرفة إذا كانت توجد أخطاء فى البرنامج.

```
Compiling Program Block...
MAIN (OB1)
Network 1, row 1, col 3: ERROR 32: (operand 1) Illegal syntax for the instruction operand.
SBR_0 (SBR0)
INT_0 (INT0)
Block Size = 0 (bytes), 1 errors

Compiling Data Block...
Block Size = 0 (bytes), 0 errors

Compiling System Block...
Compiled Block with 0 errors, 0 warnings

Total Errors: 1

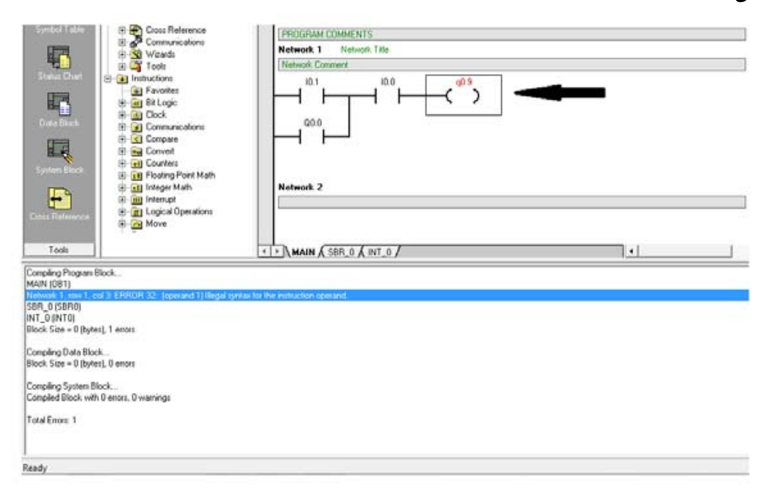
Ready
```

ملاحظة:

بعد الضغط على compile تظهر رسالة تحتوى على العدد الكلى للأخطاء بل وتحتوى أيضاً على رسالة توضيحية خاصة بكل خطاء تشير إذا كان الخطاء هو بسبب مسمى غير صحيح, خطاء فى الرسم, عدم اكتمال الرسم, وجود قطع فى البرنامج, قد تم عمل قفله على مفتاح معين وهكذا...

حيث تقوم وحدة البرمجة بتحديد فى أى فرع يوجد الخطاء ومن ثم تقوم بتحديد فى أى سطر يوجد الخطاء وفى النهاية تقوم بتحديد فى أى عمود أى أنه يقوم بتحديد مكان السبب الرئيسى للخطاء بالتفصيل.

ثالثاً: يتم الضغط مرتين على الرسالة التى تدل على وجود خطاء حيث ستشير وحدة البرمجة على السبب الرئيسى فى العطل تلقائياً.



ملاحظة:

في حاله عدم فهم الرسالة الخاصة بالمشكلة يكفى الضغط على الرسالة مرتين وسينتقل المؤشر إلى مكان المشكلة فى البرنامج تلقائياً.

النوع الثانى:

وهى ثانى أكثر الأنواع التى يتم التعرض لها داخل الـ PLC وتسمى بالـ non fatal errors حيث قد يجتاز عدد الأعطال من هذا النوع العشرات ولكن لا يبدو الموضوع صعب أو معقد لأن وحدة البرمجة تقوم بمساعدتك فى اكتشاف وحل العطل, بعض الأعطال يمكن اكتشافها قبل حتى أن يتم نقل البرنامج إلى جهاز الـ PLC والبعض يتم اكتشافها بعد تشغيل وحدة البرمجة ولكنها لا تتسبب قط بتوقف زحدة البرمجة كما سنوضح فى الأمثلة التالية.

حيث أنه حين يتم البدء فى نقل البرنامج من الكمبيوتر إلى الـ PLC يقوم البرنامج بمراجعة التمرين المرسوم للتأكد من عدم وجود إى أعطال, إما فى حالة وجود أعطال فسيساعدك البرنامج على اكتشاف الأعطال المساعدة المقدمة من قبل وحدة البرمجة للمساعدة فى اكتشاف وحل النوع الثانى من الأعطال تتمثل فى نقطة واحدة فقط:

- فأن البرنامج يقوم تلقائياً بتحديد مكان العطل بل ويقترح عليك الحل إيضاً.

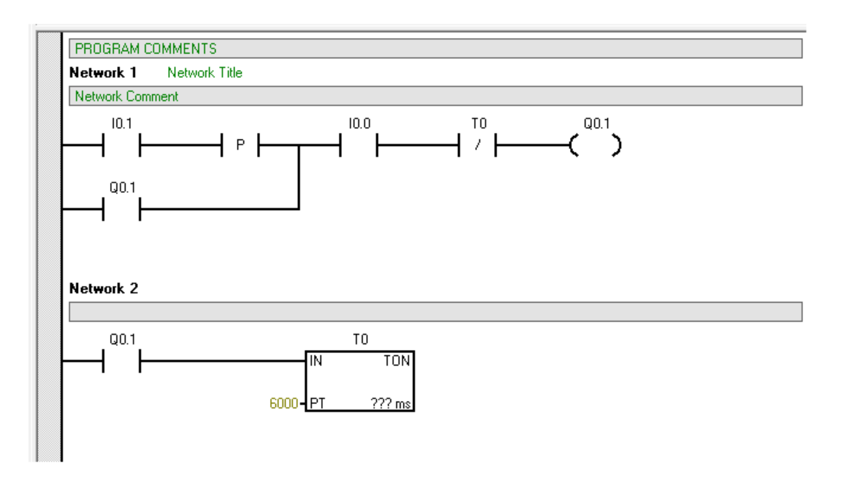
الأخطاء غير القاتلة التى تكتشف من قبل وحدة البرمجة يمكن أن تكتشف قبل أو بعد تحميل البرنامج.

الأول (الأخطاء التى تكتشف قبل تحميل البرنامج).

مثال على الأخطاء التى تكتشف قبل تحميل البرنامج لمحرك يعمل يدوياً ويقف بعد زمن من التشغيل.

من المؤكد أن هذا النوع من الأخطاء ليس متعلق بعملية حسابية تستدعى عمل وحدة البرمجة لكى يتم التأكد من سلامتها أو من أنها تعمل بشكل سليم بل من المؤكد أن الأخطاء التى تكتشف قبل تحميل البرنامج هى أخطاء مرطبته بمسميات فقط.

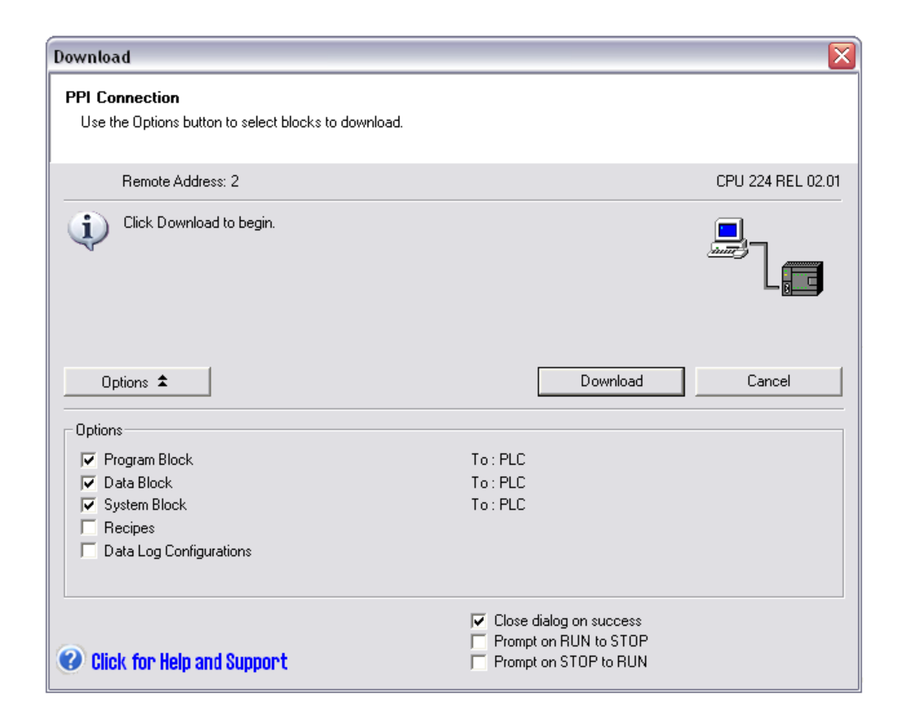
أولاً: يتم رسم البرنامج باستخدام أى لغة من لغات البرمجة.



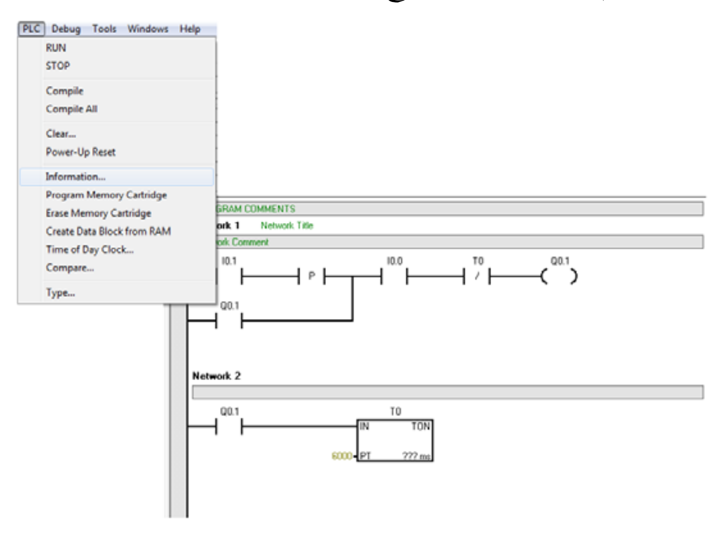
ثانياً: يتم الضغط على أمر compile لمعرفة إذا كانت توجد أخطاء فى البرنامج.



ثالثاً: كالعادة يتم الضغط بعد ذالك على أمر download لتحميل البرنامج على وحدة البرمجة ولكن ستظهر رسالة تدل على وجود خطاء دون أن تشير وحدة البرمجة إلى السبب الرئيسى فى المشكلة.



رابعاً: يتم الضغط على قائمة PLC ومن ثم يتم اختيار صفحة information لمعرفة كم عدد الأخطاء فى البرنامج, فى أى فرع توجد وما سبب الأخطاء بل وستشير وحدة البرمجة إلى رقم العطل حيث يوجد فى صفحة help تسلسل لأرقام الأعطال كما سنوضح بعد قليل.



خامساً: يتم العودة إلى البرنامج الرئيسى مرة أخرى ومن ثم يتم تفحص الأفرع التى تحتوى على أعطال, فمثلاً فى هذا المثال المشكلة هى بسبب استخدام أسم غير صحيح مع المؤقت الزمنى.

ملاحظة:

لمعرفة الأسماء الممكن استخدامها مع المؤقتات الزمنية أنظر صفحة ١٥٢ بالجزء الأول من كتاب برمجة التحكم المنطقى PLC.

الثانى (الأخطاء التى تكتشف بعد تحميل البرنامج).

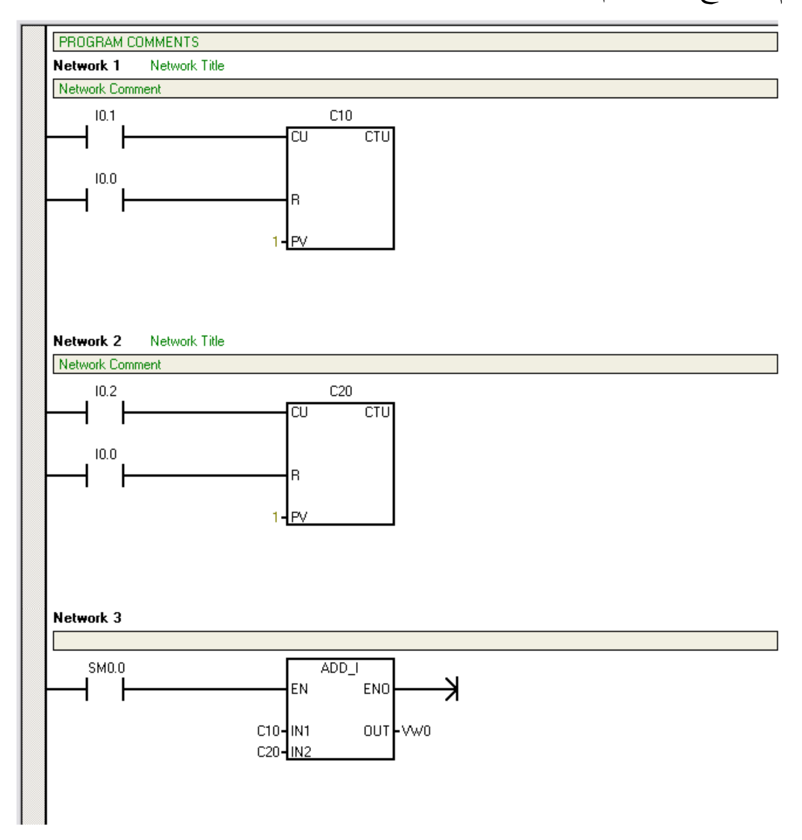
مثال على الأخطاء التى تكتشف بعد تحميل البرنامج يقوم بجمع المنتج الخاص بخطين أنتاج يقوم بتصنيع نفس المنتج.

بعض مشاكل الـ Hardware المكتشفة مثل انخفاض جهد بطارية الذاكرة الاحتياطية تسبب خطأً من النوع غير القاتل بل يمكن أن تنتج هذه الأخطاء أيضاً بسبب أن نتيجة العملية الحسابية أكبر بكثير من أن يمكن حفظها في عنوان الذاكرة المختار.

من المؤكد أن هذا النوع من الأخطاء ليس متعلق بمسميات يمكن اكتشافها دون اللجوء إلى وحدة البرمجة بل من المؤكد أن الأخطاء التى تكتشف قبل تحميل البرنامج هى أخطاء مرطبته بعملية حسابية تستدعى عمل وحدة البرمجة لكى يتم التأكد من سلامتها أو من أنها تعمل بشكل سليم فقط.

برنامج لجمع قيمة العداد C10 و C20.

أولاً: يتم رسم البرنامج باستخدام أى من لغات البرمجة.



ثانياً: يتم الضغط على أمر compile لمعرفة إذا كانت توجد أخطاء فى البرنامج.

Compiling Program Block...
MAIN (OB1)
SBR_0 (SBR0)
INT_0 (INT0)
Block Size = 35 (bytes), 0 errors

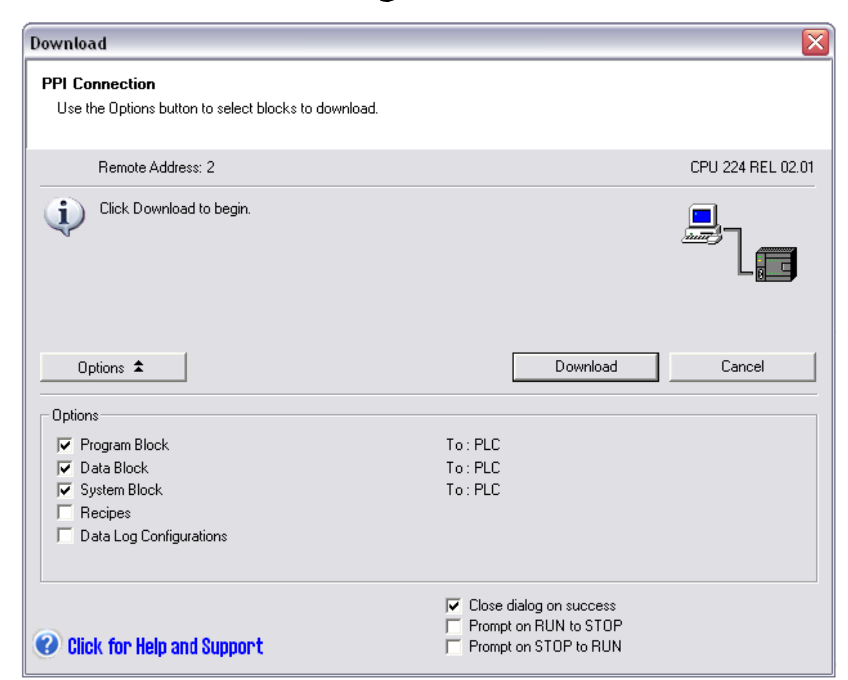
Compiling Data Block...
Block Size = 0 (bytes), 0 errors

Compiling System Block...
Compiled Block with 0 errors, 0 warnings

Total Errors: 0

Ready

ثالثاً: يتم الضغط على أمر download لتحميل البرنامج على وحدة البرمجة.



ملاحظة:

سوف يعمل البرنامج بشكل ممتاز حيث ستقوم وحدة البرمجة بجمع قيمة العداد C10 والعداد C20 معاً وتقوم بتحميل الناتج على المتغير VW0, ولكن ستكون المشكلة عندما يصبح مجموع قيمة العدادين معاً أكبر من قيمة المتغير VW0 فتشير وحدة البرمجة إلى حدوث خطاء من النوع "غير القاتل" بعد تحميل البرنامج.

النوع الثالث:

وهى أقل الأنواع التى يتم التعرض لها داخل الـ PLC وتسمى بالـ fatal errors حيث لا يجتاز عدد الأعطال من هذا النوع الخمس عشر نوع ولكن يعتبر هذا هو أصعب أنواع الأعطال لأن وحدة البرمجة لا تقوم بمساعدتك فى تحديد مكان العطل أو من الذى تسبب فى حدوثه, حتى أن جهاز البرمجة لا يلاحظ العطل قبل حتى أن يتم نقل البرنامج إلى وحدة البرمجة ولكن يحدث العطل عندما يتم التعرض إلى هذا الجزء من البرنامج فعلياً كما فى المثال التالى.

حيث أنه حين يتم البدء فى نقل البرنامج من الكمبيوتر إلى وحدة البرمجة يقوم البرنامج بمراجعة التمرين المرسوم للتأكد من عدم وجود إى أعطال ولكنه لن يتمكن من اكتشاف هذا النوع المحدد من الأعطال ولكن فى حالة حدوث العطل فيما بعد فسيساعدك البرنامج على اكتشاف الأعطال.

المساعدة المقدمة من قبل وحدة البرمجة للمساعدة فى اكتشاف وحل النوع الثالث من الأعطال تتمثل فى نقطة واحدة:

- فأن البرنامج يقوم تلقائياً بأظهار رسالة تشير إلى وجود عطل ولكن دون تحديد المكان أو من المتسبب فى العطل.

الأخطاء القاتلة Fatal Errors وهي التي تسبب أن تترك وحدة البرمجة نمط الـ Run وتصبح في نمط الأخطاء Fault. يمكن أن يكون سبب الخطأ القاتل هو اكتشاف مكونات من الـPLC لا تعمل عند إجراء الفحص الذاتي في البداية أو عند استخدام هذه المكونات أثناء تنفيذ البرنامج .

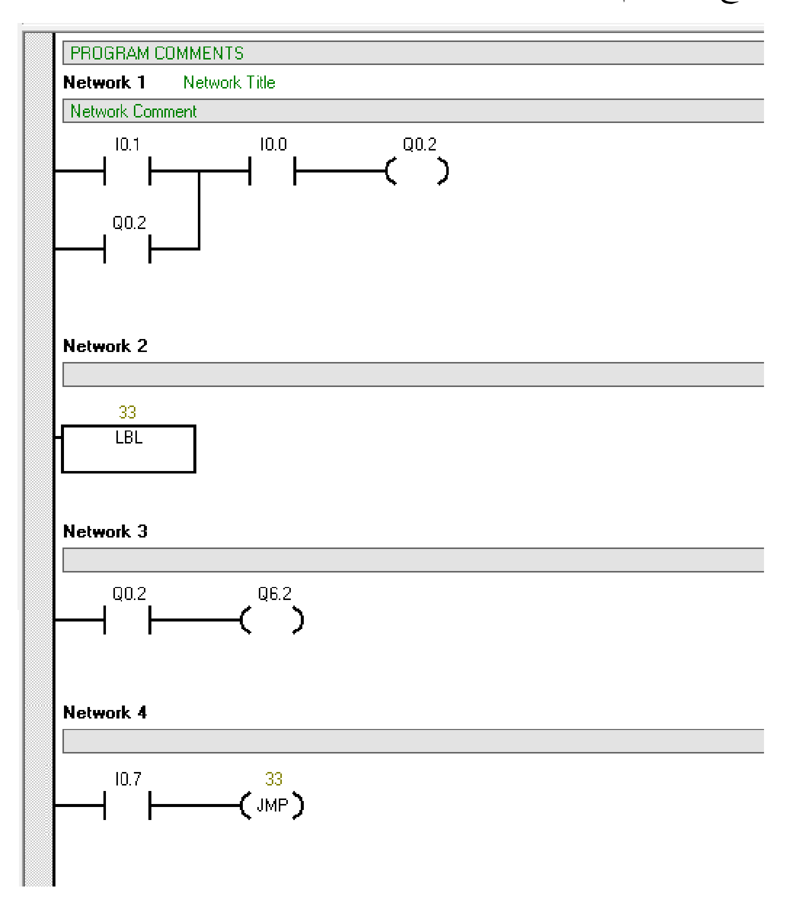
بعض المشاكل البرمجية أو المشاكل في الإعدادات (مثل انتهاء مؤقت مراقبة من نوع Watch Dog) هي أيضاً أسباب للأخطاء القاتلة.

عندما تدخل وحدة البرمجة نمط الـ Fault فإنها تضيء ديود ضوئي خاص بالأخطاء وتضع جميع المخارج في حالة Off (أو تجمد بعضها عند آخر حالة كانت عليها). وتقوم وحدة البرمجة أيضاً بتخزين رمز العطل Fault Code في الذاكرة. يستطيع المبرمج أن يقرأ رمز الخطأ ليحدد سبب العطل ويحل المشكلة ثم يصفر حالة الخطأ أو يعيد تشغيل وحدة البرمجة ليدخلها في حالة الـ Run.

وحدة البرمجة الحديثة تستطيع الاحتفاظ بوصف مفصل لبعض الأخطاء الحاصلة مؤخراُ وتسمى بالـ history.

مثال على النوع الثالث لتمرين يحتوى على عملية القفز.

أولاً: يتم رسم البرنامج باستخدام أى من لغات البرمجة.



ثانياً: يتم الضغط على أمر compile لمعرفة إذا كانت توجد أخطاء فى البرنامج.

```
Compiling Program Block...
MAIN (OB1)
SBR_0 (SBR0)
INT_0 (INT0)
Block Size = 35 (bytes), 0 errors

Compiling Data Block...
Block Size = 0 (bytes), 0 errors

Compiling System Block...
Compiled Block with 0 errors, 0 warnings

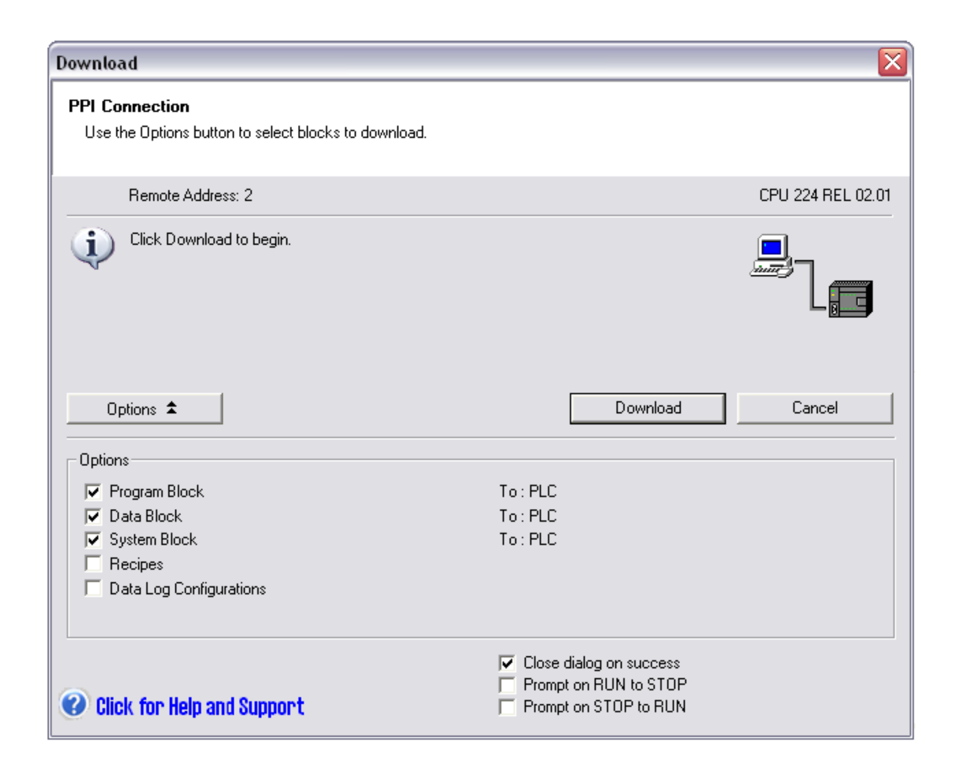
Total Errors: 0

Ready
```

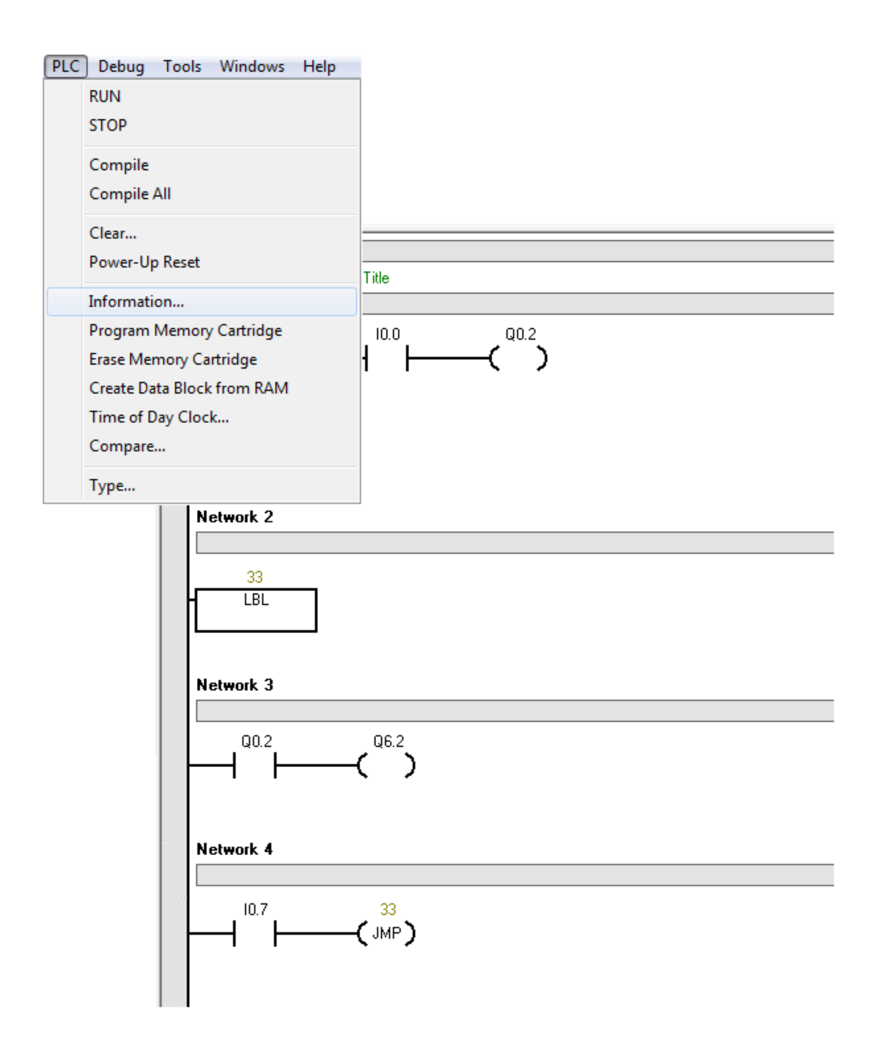
ملاحظة:

عند الضغط على أمر compile لا يدل جهاز البرمجة الـ PLC بأى طريقة على وجود مشكلة.

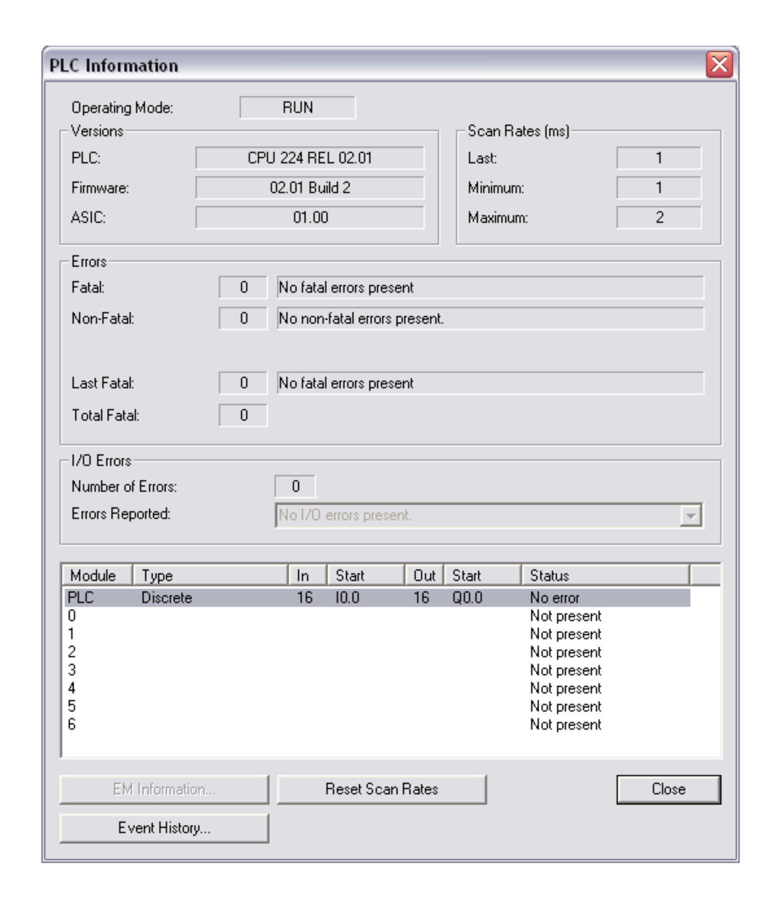
ثالثاً: كالعادة يتم الضغط بعد ذالك على أمر download لتحميل البرنامج على وحدة البرمجة وكأمر طبيعي لن ستظهر أي رسالة تدل على وجود خطاء أو عطل أو مشكلة.



رابعاً: يتم الضغط على قائمة PLC ومن ثم يتم اختيار صفحة information لقراءة رسالة توضيح حيث تشرح نوع المشكلة التي في البرنامج, ولكن بدون تحديد في أي فرع توجد المشكلة أو ما سبب الخطاء ولكن وستشير وحدة البرمجة إلى رقم العطل حيث يوجد في صفحة help تسلسل لأرقام الأعطال كما سنوضح بعد قليل.



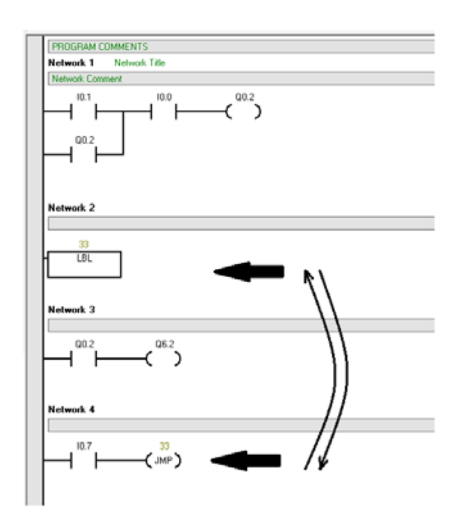
باستخدام هذه الصفحة يتمكن على المستخدم معرفة تفصيل متعددة متعلقة بنوع وحدة البرمجة, عدد المدخلات, المخراجات, عدد وأنواع الوحدات الأضافية.



ملاحظة:

- جهاز البرمجة الـ PLC لا يدل بطريقة مباشرة على سبب المشكلة ولكنة يقوم بإظهار رسالة توضيحية تساعد فى التعرف على سبب المشكلة ومن ثم تساعد فى التوصل إلى الحل.
- هذا هو النوع الأسوأ بالنسبة للأعطال الممكن التعرض لها أثناء البرمجة حيث لا تشير وحدة البرمجة نهائياً بوجود عطل إلى أن يحدث ولكن كبداية للتفاؤل لا توجد أعطال كثيرة من هذا النوع فلا يتعدى عددهم العشرين عطل مقارنتاً بالأنواع الأخرى التى قد تحتوى على عدة مئات من الأعطال.

خامساً: يتم العودة إلى البرنامج الرئيسى مرة أخرى ومن ثم يتم تفحص الأفرع التى تحتوى على أعطال, فمثلاً فى هذا المثال المشكلة هى بسبب استخدام أمر القفز بطريقة معكوسة.



ملاحظة:

- عند حدوث هذا النوع من الأعطال تضاء لمبة system fault باللون الأحمر ولمبة stop باللون الأصفر لكى تشير إلى أنه بسبب عطل ما أدى هذا إلى توقف وحدة البرمجة, لمعرفة تفاصيل أكثر عن لمبات الإشارة أنظر صفحة ٤٥ بالجزء الأول من كتاب برمجة التحكم المنطقى PLC.
- بفصل الكهرباء عن وحدة البرمجة ثم أعادة تشغيلها مرة أخرى يختفى العطل بطريقة مؤقتة لأن المشكلة لم تحل بعد فماذال البرنامج كما هو للأننا لم نقم سوا بأمر restart.
- لمعرفة الطريقة الصحيحة لاستخدام أمر القفز أنظر صفحة ١٥٢ بالجزء الثانى من كتاب برمجة التحكم المنطقى PLC.

صفحات المساعدة:

بالرجوع إلى صفحات help يمكن حصر جميع أنواع المشاكل ال errors الممكن حدوثها فى البرنامج وسيتم شرح بعضها فى الجدول التالى.

م	الاسم	الشرح
١	Error 46	انقطاع بالدائرة.
٢	Error 45	قفله بالدائرة.
٣	Error 32	خطاء بالكتابة.
٤	Error 47	خطاء بفرع البرمجة.
٥	Error 48	خطاء فى مفتاح الـ positive أو الـ negative.
٦	Error 57	خطاء بالقيمة.
٧	Error 2779	خطاء بالرموز.
٨	Error 33	رمز غير معرف بالبرنامج.
٩	Error 36	خطاء بنوع أو بحجم البيانات المستخدمة
١٠	Error 1	لا يتوافق نوع الوحدة مع استخدام الـ ENO.
١١	Error 11	لا يتوافق نوع الوحدة مع عملية معينة فى البرنامج.
١٢	Error 13	لا يتوافق نوع الوحدة مع استخدام الـ END.
١٣	Error 14	لا يتوافق نوع الوحدة مع استخدام الـ Subroutine.

١٤	Error 38	توجد عملية عير مكتملة بعد.
١٥	Error 43	مشكلة بسبب المداخل والمخارج التناظرية.
١٦	Error 49	مشكلة بسبب العداد.
١٧	Error 52	لا يمكن للخرج إن يسبق الدخل.
١٨	Error 53	لا يمكن توصيل أمر END دون مفتاح للتحكم فيه.
١٩	Error 54	لم يتم توصيل الخرج بإحدى الفروع.
٢٠	Error 2700	خطاء فى عملية بلغة الـ STL.
٢١	Error 80	حجم البرنامج أكبر من حجم الذاكرة.
٢٢	Error 300	خطاء بطرف التوصيل بوحدة البرمجة.
٢٣	Error 301	وضعية خطاء لمفتاح ضبط الحالة الـ mode switch.
٢٤	Error 306	لا يمكن تنفيذ العملية بسبب وجود كلمة مرور.
٢٥	Error 313	مشكلة بسبب صغر حجم ذاكرة الوحدة.
٢٦	Error 327	خطاء أثناء إرسال بعض البيانات إلى وحدة البرمجة.
٢٧	Error 328	خطاء أثناء استقبال بعض البيانات من وحدة البرمجة.
٢٨	Error 329	خطاء أثناء التوصيل بالوحدة تأكد من رقم الوحدة وسرعة الاتصال و الكبل.
٢٩	Error 2770	لقد سبق ذكر نفس العنوان داخل صفحة
٣٠	Error 2771	خطاء بالعنوان المذكور بصفحة الـ Data Block
٣١	Error 2772	خطاء بالرمز المذكور بصفحة الـ Data Block

٣٢	Error 2773	خطاء بالقيمة لأنها أكبر من حجم الذاكرة بصفحة الـ Data Block
٣٣	Error 2774	خطاء بالكتابة لأنها أكبر من الحد المسموح به بصفحة الـ Data Block

صفحة الريليهات الخاصة:

بالرجوع إلى صفحات الريليهات الخاصة المتاحة ضمن صفحة الرموز التى تم شرحها بالتفصيل فى الجزء الثانى من كتاب برمجة التحكم المنطقى .P.L.C وسيتم شرح بعضها فى الجدول التالى.

م	الاسم	
١	SM0.0	يغلق بطريقة دائمة
٢	SM0.1	يغلق لزمن دورة واحدة ثم يفتح مرة أخرى.
٣	SM0.2	يغلق فى حالة فقدان أى قيمة مسجلة بوحدة البرمجة.
٤	SM0.3	يغلق فى حالة تشغيل الوحدة على وضع RUN.
٥	SM0.4	يغلق لثلاثين ثانية ويفتح لثلاثين ثانية أخرى.
٦	SM0.5	يغلق لنصف ثانية ويفتح لنصف ثانية أخرى.
٧	SM0.6	يغلق لزمن cycle ويفتح لزمن cycle أخر.
٨	SM0.7	يغلق عندما يكون المفتاح فى وضع run و يفتح عندما يكون فى وضع term.
٩	SM1.0	يغلق عندما يكون الناتج الخاص بأى عملية حسابية يساوى صفر.
١٠	SM1.1	يغلق عندما يكون الناتج أكبر من حجم الذاكرة المستخدمة.
١١	SM1.2	يغلق عندما يكون الناتج الخاص بأى عملية حسابية سالب.
١٢	SM1.3	يغلق عندما يتم القسمة على صفر فى أى عملية حسابية.

١٣	SM4.7	يغلق عندما يتم تطبيق أمر Force على أى عنوان.
١٤	SM5.0	يغلق عند حدوث أى عطل بخصوص وحدة المداخل أو المخارج.
١٥	SM1.6	يغلق عندما يتم تحويل رقم BCD غير صحيح إلى Binary.

طريقة أخرى لتحديد الفرع الذى يحتوى على العطل:

توجد طريقة أخرى يمكن بواسطتها التعرف على مكان المشكلة, هى طريق غير مباشرة حيث تشير إلى الفرع الذى بة المشكلة دون تحديد السبب الرئيسى للمشكلة.

الشرح.

لقد تم التوضيح فى الكتابين السابقين على أنه توجد عدة لغات للبرمجة بل ويمكن تغير لغة البرمجة بعد تنفيذ البرنامج أيضاً, ولذالك فأنة من المهم أن نشير إلى أمر ذات غاية فى الأهمية وهو أنه يمكن تحديد الفرع الذى يوجد به مشكلة بواسطة التحويل من لغة إلى لغة أخرى حيث ستقوم وحدة البرمجة بتحويل جميع الأفرع ماعدا الأفرع التى بها مشاكل حيث ستظهر مقابلها كلمة "invalid" أى أن وحدة البرمجة لم تستطيع فهم هذه الأفرع ومن هنا يمكن معرفة الأفرع التى يوجد بها مشاكل وبالعودة إلى اللغة الأصلية مرة أخرى يمكن تفحص أفرع البرمجة التى لم يتم تحويلها حيث من المؤكد أنها تحتوى على أوامر أو عناوين غير صحيحة.

تمرين للتوضيح

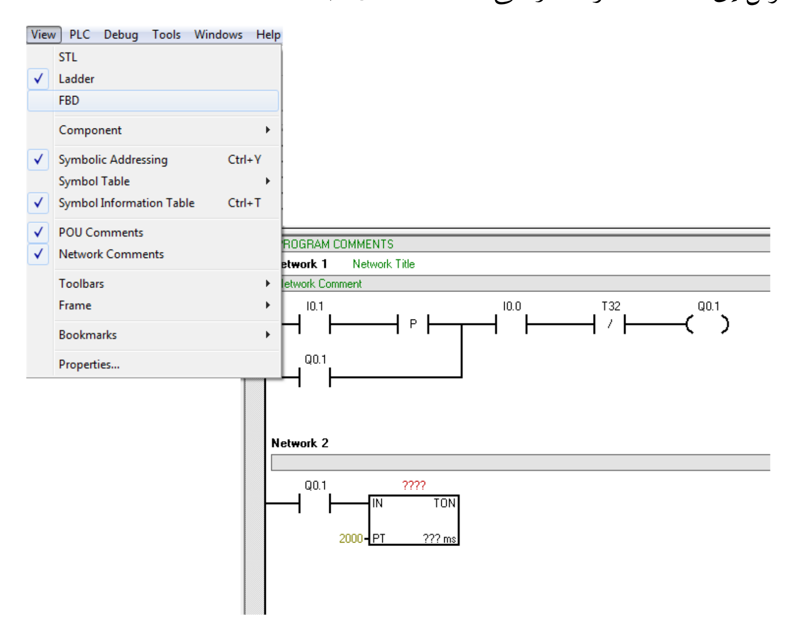
برنامج لمحرك يعمل يدوياً ولكن يقف أوتوماتيكياً بعد ثنيتان من زمن التشغيل.

الخطوات:

أولاً: يتم رسم البرنامج باستخدام أى لغة.

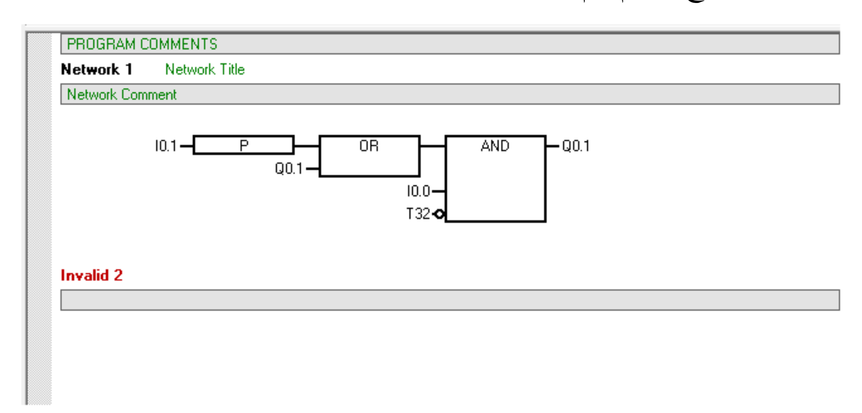
PROGRAM COMMENTS
Network 1 Network Title
Network Comment
I0.1
P
I0.0
T32
/
Q0.1
Q0.1
Network 2
Q0.1
????
IN
TON
2000
PT
??? ms

ثانياً: يتم التحويل إلى أى لغة أخرى غير التي استخدمت في البداية.

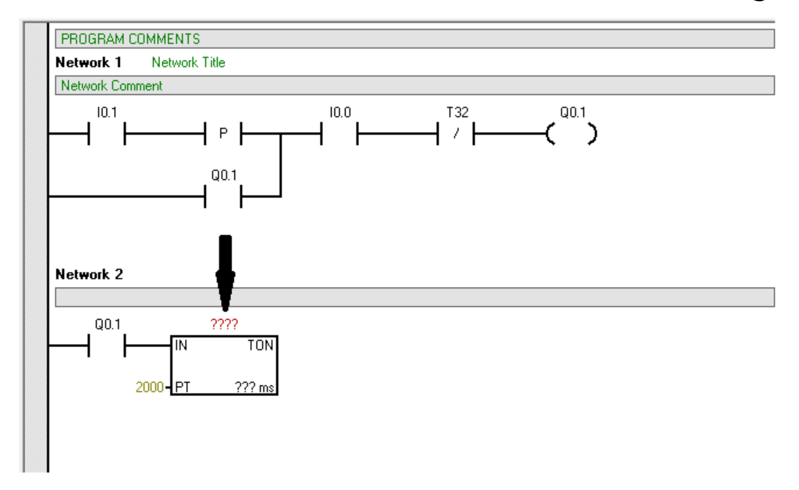




ثالثاً: يتم البحث عن الأفرع التى لم يتم تحويلها إلى اللغة الجديدة.



رابعاً: يتم الرجوع مرة أخرى إلى اللغة الأصلية لمعرفة السبب الأساسى فى العطل.



ملاحظة:

- نظراً إلى أن هذه الطريقة تشير إلى الفرع الذى بة المشكلة دون تحديد السبب الرئيسى للمشكلة لذالك يفضل دائماً عدم رسم الكثير من الأوامر فى نفس الفرع لسهولة القراءة والفهم فيما بعد وخاصة فى حالة وجود أعطال.
- هذه الطريقة تشير إلى الأخطاء الكتابية فقط (أسم دخل خطاء – أسم خرج خطاء – أسم ريليه خطاء – أسم عداد خطاء – أسم مؤقت زمنى خطاء).
- فى حاله عدم استخدام نقاط الحماية بالنسبة لمحرك عكس حركة فهذا لا يعنى أى شىء لوحدة البرمجة لأن كيف لوحدة البرمجة أن تعرف من الأساس أننى أقوم بتصميم محرك عكس حركة.

إصلاح الـ Hardware خارج وحدة البرمجة:

تحتوي جميع الـ PLC's على ديودات ضوئية في وحدة المعالج CPU وعلى وحدات الدخل الخرج وأحياناً على وحدة التغذية.
بشكل عام دائماً فى وحدة البرمجة يشير الضوء الأحمر إلى وجود مشكلة والضوء الأخضر يعني Ok . إذا كان أحد الديودات الضوئية يومض بالون الأخضر فإنه عادةً يعني أن وظيفةً ما تعمل.
إن الترجمة الصحيحة للحالة التي تشير إليها الديودات الضوئية يمكن أن توفر عليك الكثير من وقت الإصلاح.
انظر إلى دليل الإصلاح الذي يوضح معاني إضاءة الديودات الضوئية في كتالوج الشركة المصنعه

الحالة الأولى: إذا كانت الـ PLC لا تدخل نمط الـRUN يمكنك أن تفحص لمعرفة إذا كانت المشكلة فى البرنامج أم في الـ Hardware :

مشكلة HARDWARE ناتجة عن مشكلة بالدخل:
إذا ظهر أن وحدة البرمجة لا تستلم إشارة من حساس ما يمكنك التأكد من ذلك بمراقبة ذاكرة صورة الدخل PII (Process Image of Inputs) بينما تشغل الحساس, راقب الديودات الضوئية على وحدة دخل الـPLC لترى إذا كانت تغير وضعها عندما يعمل الحساس المناسب:

أ- إذا كانت الديودات لا تتغير استخدم مقياس أفو للتأكد أن الإشارة عند دخل وحدة البرمجة تغير حالتها, إذا لم تكن كذلك افصل الحساس عن وحدة البرمجة وافحص عمله بشكل مستقل.
ابحث عن مشاكل وحدات الدخل الخارجية أو طريقة توصيل تغذية المدخلات, تذكر أن بعض وحدات الدخل التى تعمل بالتيار المستمر في بعض وحدات البرمجة قد تكون من النوع البالع للتيار Current Sinking أو تسمى أيضاً باسم PNP وبالتالي فإنه يجب على الحساس أن يوصل طرف التغذية الموجب إلى وحدة الدخل بوحدة البرمجة والبعض الآخر من وحدات البرمجة قد تكون من النوع المعطي للتيار Current Sourcing أو تسمى أيضاً باسم NPN وبالتالي فإنه يجب على الحساس أن يوصل طرف التغذية السالب إلى وحدة الدخل بوحدة البرمجة.

ب– إذا تغيرت الديودات الضوئية فربما يكون العطل في البرنامج ذاته, للتأكد ضع أمر "نهاية دورة المسح" END كأول سطر في برنامجك لكى يتم تجنب أى عمل للبرنامج أثناء الفحص ثم راقب صورة الدخل ثانية والبرنامج يعمل Run إذا تغيرت الخانة مع الديود إذا كانت المشكلة في البرنامج لابد أن البرنامج كان يكتب قيمة في خانة صورة الدخل.

إذا ظلت خانة صورة الدخل لا تتغير مع حالة الديود الضوئي الذي على وحدة الدخل يمكن أن يكون العطل في الحساس فقد يكون التيار الكهربى الناتج من دائرة الحساس غير كاف لتغيير حالة الدخل بالرغم من أنه يضيء الديود الضوئي استخدم الأفو فستكتشف تغير صغير بالجهد عند توصيل الدخل.

مشكلة HARDWARE ناتجة عن مشكلة بالخرج:

إذا كان أحد المشغلات Actuators (ريليه, صمام، كونتكتور..) لا يستلم إشارة من وحدة البرمجة راقب الديودات الضوئية على وحدة الخرج للتأكد أنها تغير وضعيتها عندما تغير وحدة البرمجة حالة الخرج:

أ– إذا كان الديود الضوئي يتغير استخدم مقياس الأفو للتأكد أن وحدة الخرج تعطي تغيرات في الإشارة كافية لكى تعمل المشغلات. إذا كان كذلك افصل المشغل عن الPLC وجرب عمله لوحده, انتبه إيضاً لبعض وحدات الخرج افحص هل هي معطوبة؟.

ب– إذا لم يكن الديود الضوئي يتغير افحص وحدة تغذية دائرة الخرج ووصلاتها مع وحدة الخرج.

ملاحظة.

- إذا كانت الحساسات والمشغلات تعمل بشكل صحيح عليك أن تبحث عن المشاكل في داخل وحدة البرمجة فقط, كما سبق ووضحنا فى الصفحات السابقة.
- فى حالة اكتشاف أن العطل ليس بسبب أمر خارج وحدة البرمجة بل أن سبب العطل هو مرتبط بالبرنامج ذاته يفضل أولاً طباعة البرنامج على ورق للتمكن من القراءة بسهولة والتعرف على سبب العطل ومن ثم حل العطل, لمعرفة الكثير عن الطباعة أنظر الفصل الخاص بالطباعة صفحة ١٦٣فى هذا الكتاب.

الباب الرابع

أجهزة القياس الكهربية

- الأجهزة التى تساعد فى أكتشاف المشكلة.
- جهـــــاز قياس فرق الجهد الفولت ميتر.
- جهـــــــاز قياس التيار الكهربى الأميتر.
- جهـــاز قياس المقاومة الكهربية الأوم ميتر.
- جهـــــــاز متعدد الأغراض الأفو ميتر.
- مكونات أجهـــــــــزة القياس الكهربية.
- طريقة استخدام بنســــــــــة الأمبير.
- الملصقات الرقمية على الأسلاك والأحمال.

الأجهزة التى تساعد فى اكتشاف الأعطال

تجرى عملية القياس بتقنيات متنوعة إلكترونية أو كهربائية أو ميكانيكية أو غيرها. وتعد التقنيات الكهربائية والإلكترونية أكثرها انتشاراً جميع المجالات العلمية بسبب تميزها بالقة والحساسية العاليتين وسرعة النتيجة وإمكان تنظيم المعلومات وتنسيقها وتحليلها التى تساعد كثيراً فى اكتشاف سبب العطل.

جهاز قياس فرق الجهد الفولت ميتر (voltmeter)

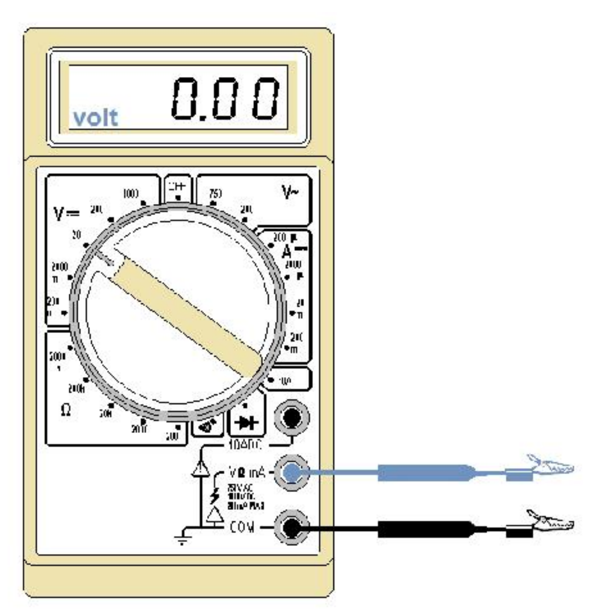
يستخدم هذا الجهاز لقياس فرق الجهد المطبق بين طرفين حمل كهربائي ما أو لقياس جهد المصدر، يوصل هذا الجهاز على التوازي مع المصدر أو الحمل الكهربائي مع شرط سريان التيار الكهربائي أي يجب أن تكون الدارة الكهربائية المراد قياس فولتيتها مغلقة، ويحتوي هذا الجهاز على مفتاح اختيار لتحديد نوع الجهد المراد قياسه هل هو متردد أم ثابت

الجهد الكهربائي جهاز يستخدم لقياس" فولت ميتر Voltmeter "، يتكون عادة من أميتر ذي ملف متحرك موصل على التوالي بمقاومة كبيرة، ونظرا لأن مقاومة الجهاز ثابتة فإن التيار الكهربائي المار في الجهاز يتناسب طرديا مع الجهد عند النقطتين اللتين يوصل بهما.

يتم تدريج . ليقيس بوحدات الفولت لمجموعة من القيم بتغير قيمة المقاومة بواسطة مفتاح اختيار. يربط جهاز الفولت ميتر على التوازي مع الدائرة الكهربائية المراد قياس جهدها.

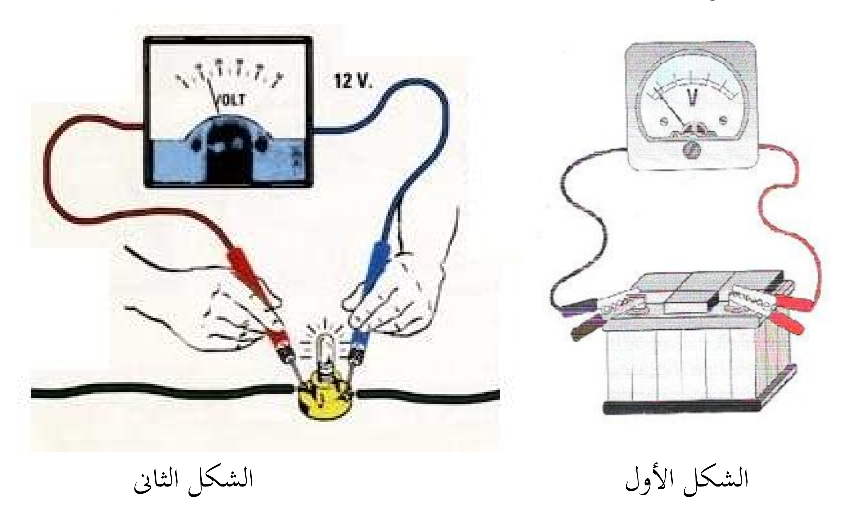
هناك أجهزة فولت ميتر ألكتروماجنيتك والكترونية تماثلي أو رقمي, لقياس الجهد المتناوب أو الجهد المستمر.

شكل الجهاز.



طريقة التوصيل:

يتم توصيل جهاز الفولت ميتر على التوازى مع المصدر المباشر للطاقة الكهربية أو على التوازى مع الحمل كما هو موضح بالشكلين التاليين.



الشكل الثانى الشكل الأول

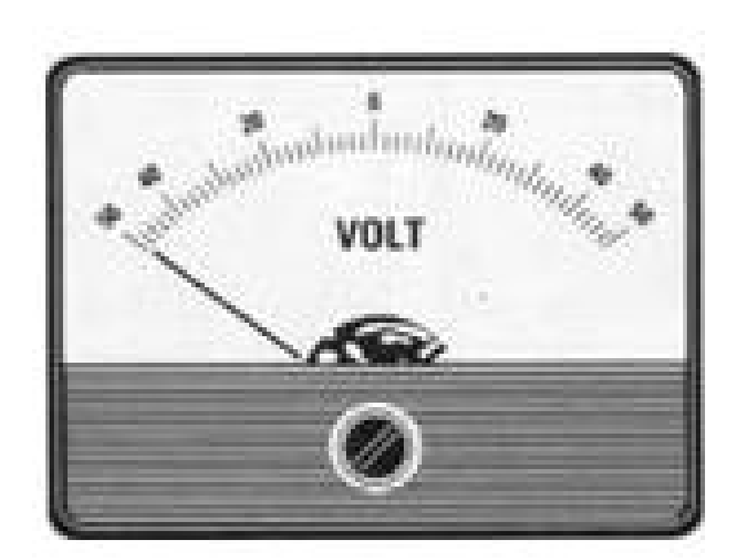
ملاحظة:

يستخدم جهاز الفولت ميتر عادتاً للتأكد من:

- تغذية وحدة البرمجة
- قيمة جهد المصدر الكهربى
- الجهد الموصل بواسطة نقطة تلامس المفتاح
- قيمة الجهد الخارج من وحدة البرمجة
- قيمة جهد الحمل ذاته

كيفية قراءة قيمة الفولت ميتر

تتم قراءة قيمة الفولت ميتر بطرق مختلفة حسب التقسيم المتبع على الشاشة ويعتمد هذا على القيم المكتوبة على الشرط الكبيرة ويعتمد أيضاً على عدد الشرط الصغيرة التى تتوسط الشرط الكبيرة



- في حالة الحاجة إلى قياس قيمة الجهد على سلك ولكن لم نتمكن من التوصيل على الطرفيين بسبب بعدهما عن بعضهما فيمكن بسهولة استخدام الجهاز التالى ويتم توصيلة على الطرف الحى بينما سيمثل جسم الإنسان الأرضى ويسمى هذا الجهاز بالـ tester.

شكل الجهاز



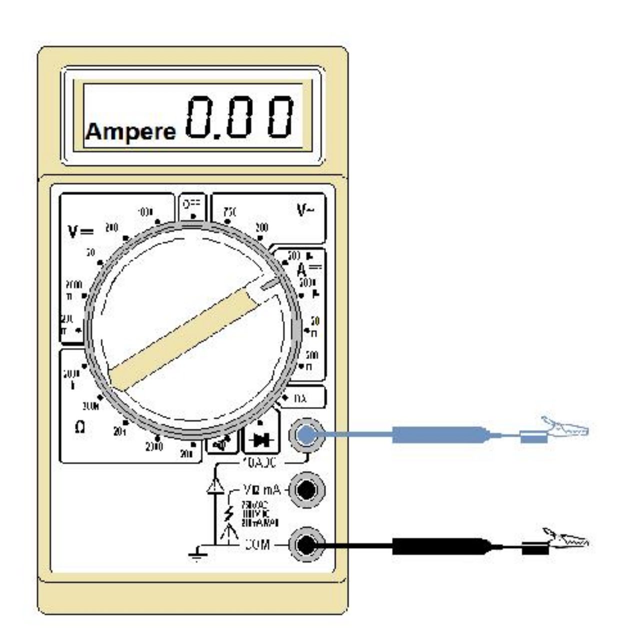
الشرح.

يستخدم جهاز التستر بنفس طريقة مفك التستر بحيث يعتمد على التلامس مع الأرضى الذى يتم بواسطة التلامس مع يد الإنسان ويحتوى الجهاز على شاشة عرض تظهر قيمة الجهد بين السلك و الأرضى.

جهاز قياس التيار الاميتر (ammeter)

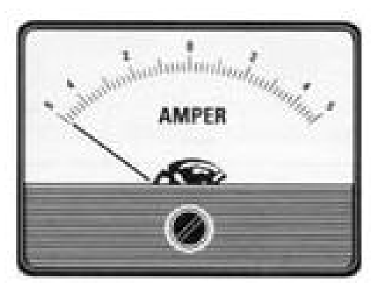
يستخدم هذا الجهاز لقياس التيار الكهربائي المار في حمل كهربائي ما، يوصل هذا الجهاز مع الحمل المراد قياس تياره على التوالي مع مراعاة ان تكون الدائرة الكهربائية مغلقة، ويحتوي هذا الجهاز على مفتاح اختيار لتحديد نوع التيار المراد قياسه هل هو متردد أم مستمر.

شكل الجهاز.



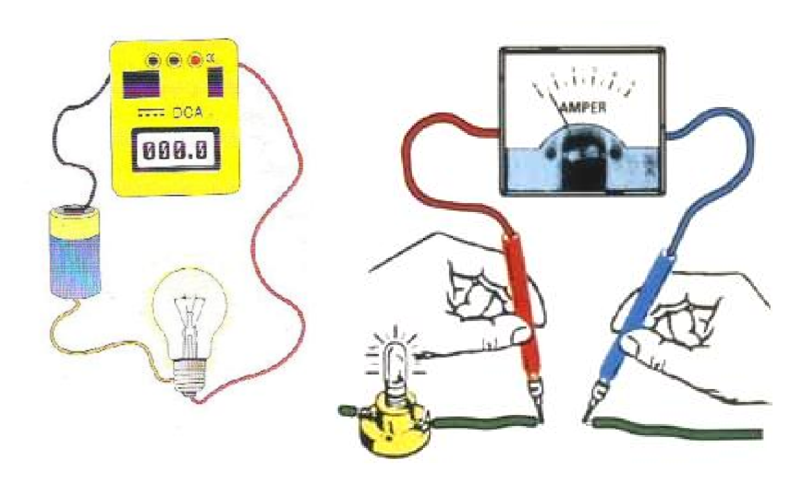
كيفية قراءة قيمة الأميتر:

تتم قراءة قيمة الأميتر بطرق مختلفة حسب التقسيم المتبع على الشاشة ويعتمد هذا على القيم المكتوبة على الشرط الكبيرة ويعتمد أيضاً على عدد الشرط الصغيرة التى تتوسط الشرط الكبيرة



طريقة التوصيل:

يتم توصيل جهاز الأميتر على التوالى مع المصدر المباشر للطاقة الكهربية أو على التوالى مع الحمل كما هو موضح بالشكلين التاليين.



ملاحظة:

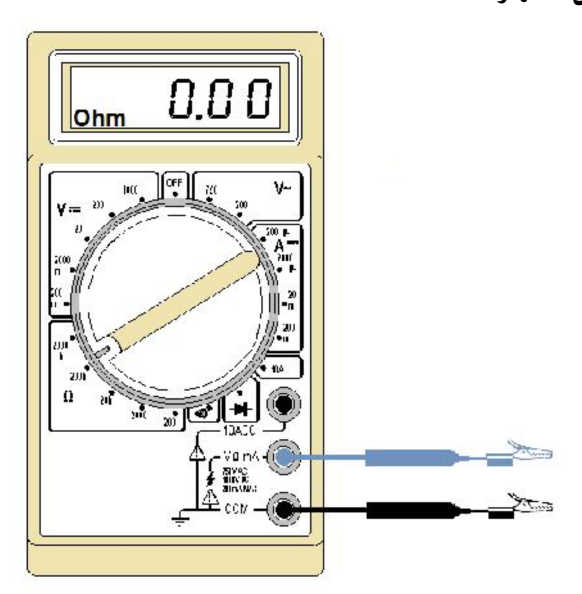
يستخدم جهاز الأميتر عادتاً للتأكد من:

- تحمل الأسلاك للتيار المسحوب
- التيار الذى يتحمله المصدر الكهربى
- قيمة التيار الخاصة بمختلف الأحمال
- لتحديد نوع الريلية لكى يتحمل قيمة التيار

جهاز قياس المقاومة الأوم ميتر (ohmmeter)

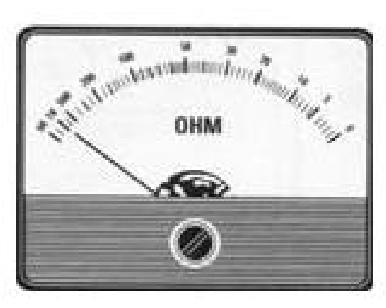
يستخدم هذا الجهاز لقياس مقاومة الأحمال الكهربائية وللتأكد من صلاحية هذه الأحمال، يوصل هذا الجهاز مع الأحمال المراد قياس مقاومتها على التوازي مع مرعاه عدم وجود سريان للتيار الكهربائي أي يجب إن تكون الدائرة مفتوحة.

شكل الجهاز.



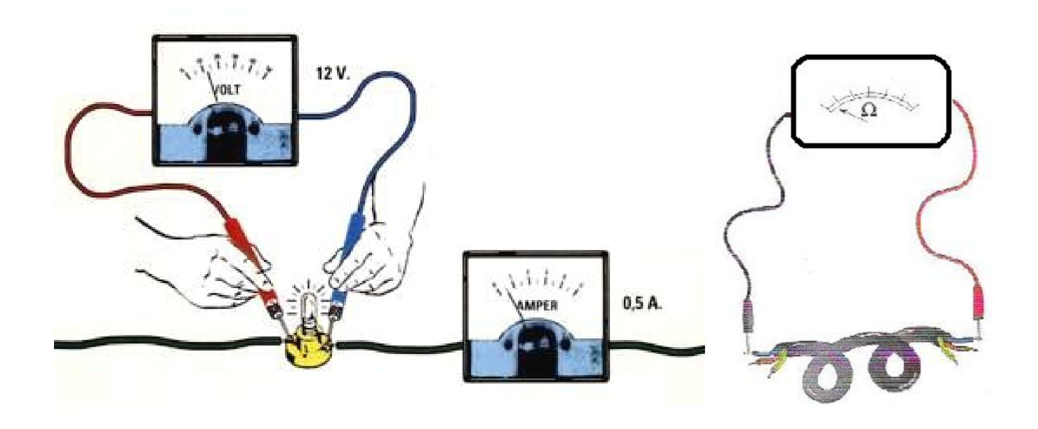
كيفية قراءة قيمة الأوم ميتر:

تتم قراءة قيمة الأوم ميتر بطرق مختلفة حسب التقسيم المتبع على الشاشة ويعتمد هذا على القيم المكتوبة على الشرط الكبيرة ويعتمد أيضاً على عدد الشرط الصغيرة التى تتوسط الشرط الكبيرة.



طريقة التوصيل:

يتم توصيل جهاز الأوم ميتر على التوازى مع الحمل كما هو موضح بالشكلين التاليين.



ملاحظة:

- جهاز القياس متعدد الأغراض الافوميتر(avometer) يجمع هذا الجهاز بين أكثر الأجهزة أهمية (الاميتر، الفولت ميتر, الاوم ميتر, ..)ويحتوي هذا الجهاز على مفتاح اختيار يمكنك من خلاله اختيار نوع الكمية المراد قياسها والتدريج المناسب
- توجد خصية أخرى تسمى بالطنان أو بالـ buzzer والتى تصدر صفارة عند اتصال طرفى الجهاز ببعضهما فى حالة أراد الشخص التأكد فقط من سلامة السلك.
- يستخدم جهاز الأوم ميتر عادتاً للتأكد من:
  - عدم انقطاع الأسلاك المستخدمة
  - لحساب قيمة مقاومة الريليهات
  - لحساب قيمة مقاومة الأحمال

شاشة قراءة القياس

تنقسم شاشات القياس إلى نوعين:

- **شاشة رقمية**

تتميز بسهولة فى قراءة القيم.

وضوح القيم المقروءة.

أمكانية تثبيت القيمة.

أضاءة خلفية للسهولة القراءة.



الشاشة الرقمية

- **شاشة تناظرية**

تتميز بدقة عالية فى القيم.



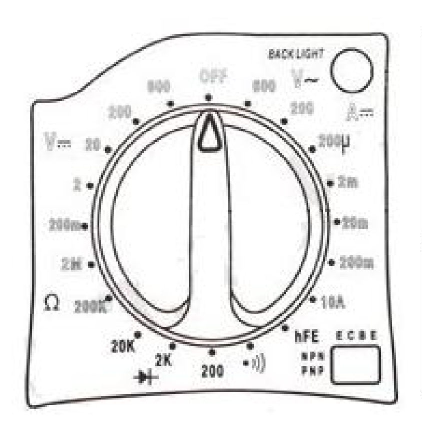
الشاشة التناظرية

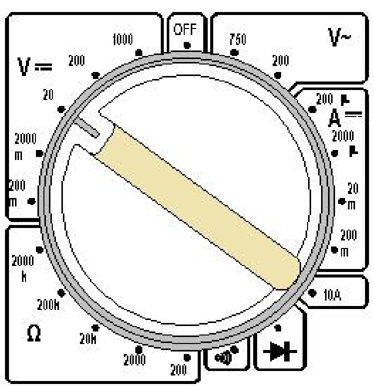
ملاحظة:

تتم قراءة القيمة مع غلق أحدى العينين للتأكد من قراءة القيمة السليمة.

مفتاح ضبط الأوضاع

يستخدم مفتاح ضبط الأوضاع للتنقل من وحدة إلى أخرى كما هو موضح.



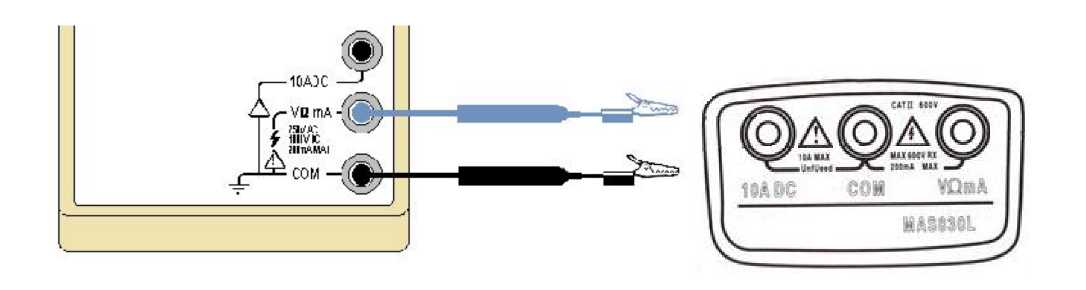


ملاحظة:

- لا يمكن ضبط المفتاح على وضع الأوم بينما يكون هناك مرور للتيار فى الدائرة.
- لا يمكن ضبط المفتاح على وضع الطنان بينما يكون هناك مرور للتيار فى الدائرة.
- من المنطقى جداً أنه عند ضبط المفتاح على وضع التيار أن يكون هناك مرور للتيار فى الدائرة.
- يتميز مفتاح الأوضاع بقيم متدرجة لنفس الوحدة فيفضل الضبط على القيمة الكبيرة فى حاله عدم معرفة مدى كبر القيمة التى سيتم قياسها.

أقطاب التوصيل

يتم نقل الأقطاب حسب نوع الوحدة المراد قياسها كما هو موضح بالشكلين التاليين.

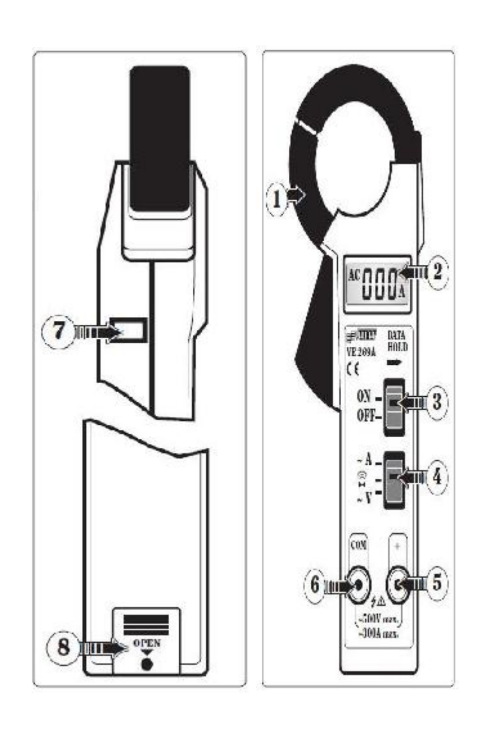


بنسة الأمبير clamp ampere

تستخدم بنسة الأمبير لقياس الأوم والتيار والفلت "المستمر والمتردد"

تعمل بنسة الأمبير بسبب المجال المغنطيسى الناتج عن مرور التيار الكهربى بالسلك.

تكوين بنسة الأمبير.



تتكون بنسة الأمبير من:

١- **الفكين** ويتم استخدامهم لمرور السلك بداخلهم.

٢- **الشاشة** وتستخدم لعرض القيم التى تم قياسها سواء بالنسبة للتيار أو الجهد أو الأوم.

٣- **مفتاح التشغيل** ويستخدم لتشغيل وفصل الجهاز.

٤- **مفتاح الضبط** ويستخدم لتغير الوضعيات المتاحة "التيار – الطنان – المقاومة – الجهد".

٥- **قطب التوصيل** ويستخدم لربط الطرف المشترك.

٦- **قطب التوصيل** ويستخدم لربط الطرف المستخدم لقياس الجهد والمقاومة.

٧- **مفتاح التثبيت** ويستخدم لتثبيت القيمة المقروءة على الشاشة.

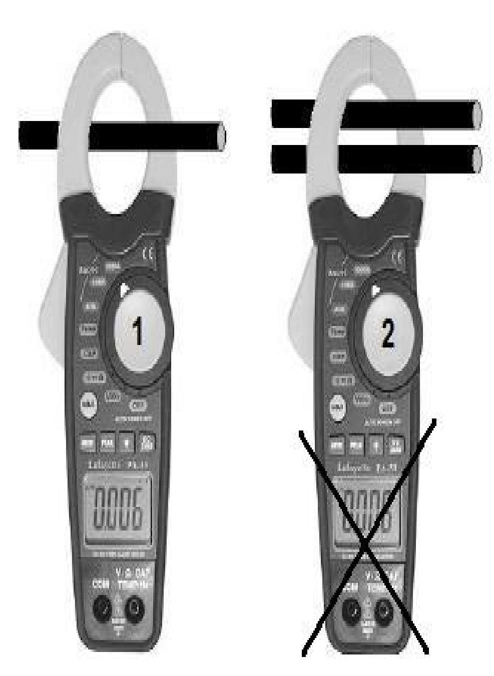
٨- **البطارية** الخاصة بنسة الأمبير.

طريقة التوصيل.

لقياس قيمة الأمبير يتم توصيل بنسة الأمبير بحيث يتم مرور أحدى الطرفين بداخل الفكين بحيث أنه يعتمد مل بنسة الأمبير على المجال المغنطيسى الناتج عن مرور التيار الكهربى بالسلك لذالك يعتبر من الخاطىء مر ر من سلك بالفكين لأن القيمة الخاصة بالمجال المغنطيسى ستتغير ومن ثم ستتغير قيمة التيار.

فمثلاً.

فى حاله مرور طرفى السلك "الكهرباء و النيوترل" معاً بين فكى بنسة الأمبير فأن القيمة الناتجة عن ذالك سوف تساوى صفر وذالك لأنة من المعروف أن قيمة التيار التى تمر بسلك الكهرباء تعادل وتساوى نفس قيمة التيار التى تمر بسلك التيوترل ولكن تمر فى الأتجاة المعاكس فلذالك سيكون المجال المغنطيسى الناتج عن التيار المار بسلك الكهرباء عكس المجال المغنطيسى الناتج عن التيار المار بسلك النيوترل فيتم تلاشى القيمتين معاً فتكون القيمة صفر.



الملصقات

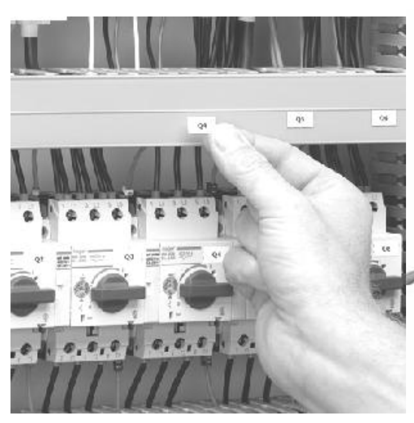


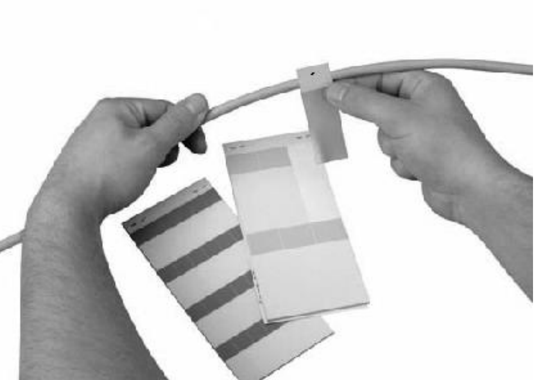
تستخدم الملصقات كوسيلة مساعدة, فمثلاً:

- للتمييذ بين الأسلاك وبعضها.
- لتوضيح أطراف وحدة البرمجة.
- لتمييز المفاتيح عن بعضها.
- لتميز الريليهات والكونتكتورات

ملاحظة:

- نظراً لأن وحدة البرمجة تثبت بعيداً عن موقع العمل لحمايتها من البيئة الصناعية المحاطة بها فلذالك يتم مد الأسلاك من المداخل والمخارج إلى وحدة البرمجة فيتم ترقيم الأسلاك لكى يتم التوصيل بسهولة.
- ترقيم الأسلاك والمفاتيح والريليهات يعتبر إساسى جداً لكى يتم الربط بين الكاتولاجات والواقع.





الباب الخامس

العوامل البيئية المؤثرة

- توصيات عامـــــــــــــــــــــــــــــة.
- العوامـــــــــــــــــــــل البيئية المؤثرة.
- السيـــــــــــــــــــــــــــاج المعدنى.
- جـــــــــــــدول المواصفات الدولية NEMA.
- جـــــــــدول الحماية الدولية ضد الأجسام الصلبة.
- التأريــض – الحمايات – الريليهات – الكونتاكتورات.
- مفاتيح حمايـــــــــــــــــــات ضد التسريب.
- البطاريـــــــــــــــــــــــــــات الـ UPS.
- مفتـــــــــــــــــــــــاح التحويل الأوتوماتيكى.
- المـــــــــــــــــــــقارنة بين أكثر من يرنامج.
- تطبيق كلمـــــــــــــــــات المرور لحماية البرنامج.
- أمثله عملية للتوضيـــــــــــــــــــــــــح.

توصيات عامة

- الاحتفاظ بنسخة مطبوعة وحاسبية عن البرامج وعن التوصيلات الكهربائية إن أمكن (قد تحتاج إليها في حال الطوارئ) ولمعرفة المزيد عن أوامر الطباعة أنظر إلى الفصل السابع الخاص بالطباعة.
- التأكد بشكل دوري من أن المداخل والمخارج مشدودة بشكل جيد.
- التأكد بشكل دوري من نظافة الكبينة (خصوصاُ المراوح والفلاتر) ووحدة التحكم من الغبار ,لأن ارتفاع الحرارة يعد من أهم العوامل المسببة للأعطال في البرامج والقطع الكهربائية.
- تأكد بشكل دوري من عدم وجود آثار للتأكسد على نقاط الربط أو فى دوائر التوصيل (خصوصاً فى الأجواء الصناعية الرطبة).
- احتفظ بقدر كافي من قطع الغيار الأكثر استخداماً مثل وحدات الدخل والخرج ... لأن كلفة التعطل لحين تأمينها أعلى من كلفة اقتناءها.
- اعمل جدول سنوي لمواعيد الصيانة الدورية وعلق نسخة منه على الآلة.

ملاحظة نهائية:

وحدات البرمجة الحديثة تخزن كميات ضخمة من معلومات يمكن الدخول إليها من قبل المستخدم لتساعده في إصلاح أعطال الـ PLC.

يمكن برمجة بعض الوحدات ببرامج خاصة للتعامل مع الأعطال تنفذ آلياً في حالة حدوث خطأ كبير قبل أن يتوقف الـ PLC عن الـ Run .

لا تغير برنامج الـ PLC لتصحح مشكلة قبل أن تتأكد أنه لا يمكن إصلاح العطل بتغيير شيء خارج وحدة البرمجة.

العوامل المؤثرة فى وحدة البرمجة

العوامل البيئية:

من المهم جداً أخذ العوامل البيئية فى الاعتبار حيث من الممكن أن تؤثر سلبياً على عمل وحدة البرمجة وقد تقصر فى العمر الأفتراضى للوحدة ومن العوامل البيئية الأساسية التى تؤثر بنسبة كبيرة على عمل وحدة البرمجة:

- **الحرارة:**

من المعروف أن أجهزة الـ PLC تعمل فى أجواء ذات حرارة بين صفر درجة مئوية و خمسين درجة مئوية, حيث فى حالة الخروج عن هذا الحد يمكن لوحدة البرمجة أن تعمل بطريقة غير سليمة ولذالك يجب تجنب قرب وحدة البرمجة من الأماكن ذات الأكثر حرارة مثل الأفران و الغلايات.

- **الرطوبة:**

من المعروف أن أجهزة الـ PLC تعمل فى أجواء ذات رطوبة قليلة بين ,فمن المعروف أن الرطوبة فى مصر قد تتعدى هذا الحد وفى حالة الخروج عن هذا الحد يمكن لوحدة البرمجة أن تعمل بطريقة غير سليمة ولذالك يجب تجنب قرب وحدة البرمجة من أماكن مكشوفة أو بالقرب من الماء.

- **التراب:**

جميع الأجهزة الإلكترونية وبالأخص تلك الحساسة تتأثر سالباً بسبب الأتربة والغبار حيث تقوم بسد فتحات التهوية المصنعة خصيصاً فى جسم وحدة البرمجة ويجب أن لا توضع وحدة البرمجة فى أماكن مكشوفة خاصاً فى المحيط الصناعى الذى قد يحتوى على الكثير من الأتربة.

- **الصداء:**

من المشاكل الأساسية التى قد تتسبب فى حدوث أعطال بوحدة البرمجة هى الأكسدة أو الصداء حيث تعمل كعازل كهربى يتراكم على نقاط التلامس الخاصة بالمداخل والمخارج بوحدة البرمجة فلذالك يجب أن لا توضع وحدة البرمجة فى أماكن مكشوفة خاصاً فى الصناعات الكيماوية التى قد تحتوى على أدخنة أكاسيد.

- **الأهتزاز:**

الاهتزازات تعد من المؤثرات السلبية على الأجهزة الإلكترونية بصفة عامه وعلى وحدة البرمجة بصفة خاصة فهى تقصر فى العمر الأفتراضى للوحدة لذالك ليس من الصحيح أن يتم تثبيت وحدة البرمجة على جسم أى معدة بل تثبت بعيدة بقدر المستطاع ويتم أيضاً وضع دعائم لكى تمتص الاهتزازات وتسمى هذه الدعائم بال "مخدات".

السياج "دولاب الحماية"

عادة يتم وضع وحدة البرمجة داخل دولاب المعدنى, حيث يتوفر الدولاب المعدنى بأبعاد وأحجام مختلفة لتتكيف مع مختلف الأجهزة وأسلاك التغذية والتحكم ويجب أيضاً بالنسبة للحجم أخذ فى الاعتبار الإضافات المستقبلية.

لقد صنعت هذه الحاويات خصيصاً لحماية ما بداخلها من الحرارة, الرطوبة, الأتربة, الصداء والاهتزازات.

الحرارة: فتتميذ هذه الحاويات بوجود مراوح تعمل أوتوماتيكياً عند ارتفاع درجة الحرارة.

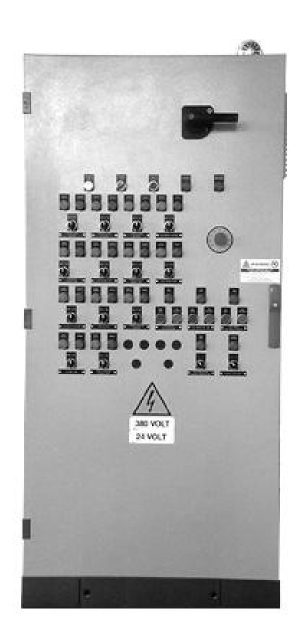
الرطوبة: تتميز بعض الحاويات بأجهزة سحب الرطوبة للحفاظ على سلامة عمل وحدة البرمجة.

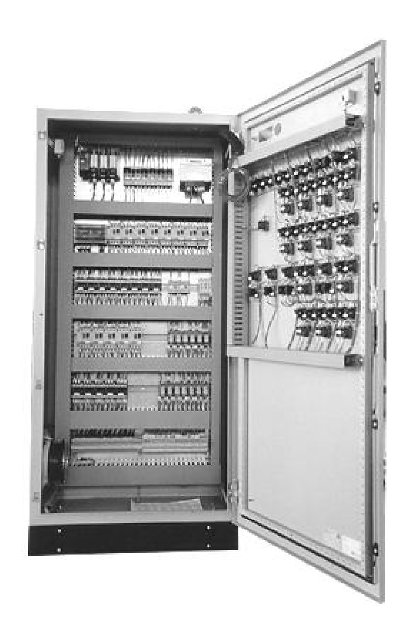
الأتربة: صنعت الحاويات بطريقة جيدة حيث تغلق بطريقة محكمة ويتم عزل جميع فتحات التهوية بفلاتر.

الصداء: تحفظ الحاويات ما بداخلها من التعرض للتأكسد بسبب طريقة العزل الجيدة.

الاهتزازات: تضمن هذه الحاويات ثبات ما بداخلها حيث يثبت الدولاب بالكامل على قواعد مرنة.

شكل الكبينة من الداخل والخارج:





ملاحظة:

- يتم تثبيت الحاويات داخل أو خارج الحائط
- يحدد حجم الحاويات بحيث يسهل تثبيت أى إضافات فى المستقبل دون الحاجة إلى تغيير الحاوية بالكامل
- يتم تصنيع الحاويات على أساس مواصفات خاصة بشروط NEMA التى تشير إلى (National Electrical Manufacturers Association) وهى جداول خاصة بتحديد بعض الثوابت الصناعية بخصوص الحمايات الكهربية.

جدول المواصفات الدولية :NEMA

IP	NEMA
IP 10	1
IP 11	2
IP 14	3R
IP 54	3
IP 54	3S
IP 56	4
IP 56	4X
IP 52	5
IP 67	6
IP 67	6P
IP 52	12
IP 52	12K
IP 54	13

- بمعرفة الكود الخاص بال NEMA يمكن بسهولة تحديد درجة الحماية ال IP سواء بالنسبة للمواد السائلة أو بالنسبة للأجسام الصلبة, كما سيتم التوضيح فى الجدولين التاليين.
- تتميز دواليب الحماية بمواصفات عالمية تسمى بال IP (International Protection) حيث يشير رقم ال IP إلى درجة عزل الدولاب ضد المواد السائلة والأجسام الصلبة كما بالجدول التالى.

جدول الحماية الدولية ضد المواد السائلة:

م	IP	الشرح
١	IP… 0	ليس معزول ضد أى مواد سائلة.
٢	IP… 1	معزول ضد سقوط المياه بطريقة عمودية.
٣	IP… 2	معزول ضد سقوط المياه بزاوية سقوط ١٥.
٤	IP… 3	معزول ضد الأمطار.
٥	IP… 4	معزول ضد رزاز المياه.
٦	IP… 5	معزول ضد رشاشات المياه.
٧	IP… 6	معزول ضد كميات من المياه.
٨	IP… 7	يمكن أن يوضع فى المياه.
٩	IP… 8	يمكن أن يغطى بالكامل تحت المياه.

جدول الحماية الدولية ضد الأجسام الصلبة:

م	IP	الشرح
١	IP 0 …	ليس معزول ضد أى أجسام صلبة.
٢	IP 1 …	معزول ضد أحجام صلبة بسمك أكبر من ٥٠ مم.
٣	IP 2 …	معزول ضد أحجام صلبة بسمك أكبر من ١٢ مم.
٤	IP 3 …	معزول ضد أحجام صلبة بسمك أكبر من ٢,٥ مم.

٥	IP 4 …	معزول ضد أحجام صلبة بسمك أكبر من ١ مم.
٦	IP 5 …	معزول ضد الأتربة.
٧	IP 6 …	معزول بالكامل ضد الأتربة.

جدول الحماية الدولية ضد التدخل البشرى:

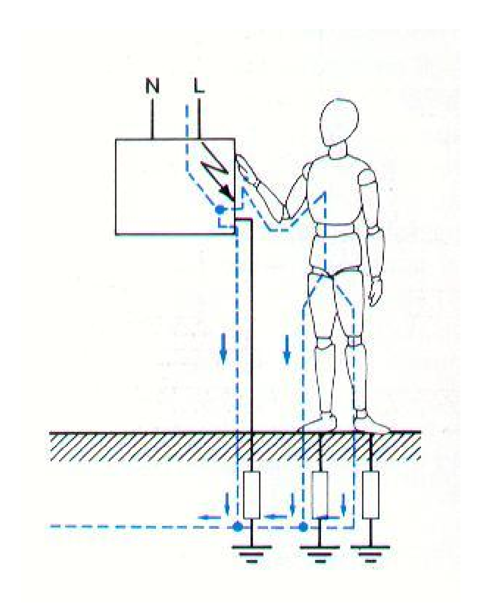
م	**IP**	الشرح
١	a	محكم ضد دخول كف اليد.
٢	b	محكم ضد دخول أصبع اليد.
٣	c	محكم ضد دخول أى عدة أو أداة.
٤	d	محكم ضد دخول أى سلك.

جدوال الحماية الدولية للمواد المستخدمة:

م	**IP**	الشرح
١	h	معدات جهد عالى.
٢	m	مختبر ضد المياه أثناء عمل الأجهزة.
٣	s	مختبر ضد المياه أثناء وقوف الأجهزة.
٤	w	يعمل تحت شروط بيئية خاصة.

تتميز الحاويات عن بعضها البعض على أساس السعة الرأسية والأفقية, بحيث تشير السعة الأفقية إلى عدد المديول الممكن توصيلها على التوالى إما السعة الرأسية تشير إلى عدد وحدات البرمجة التى يمكن تثبيتها بالحاوية.

التأريض



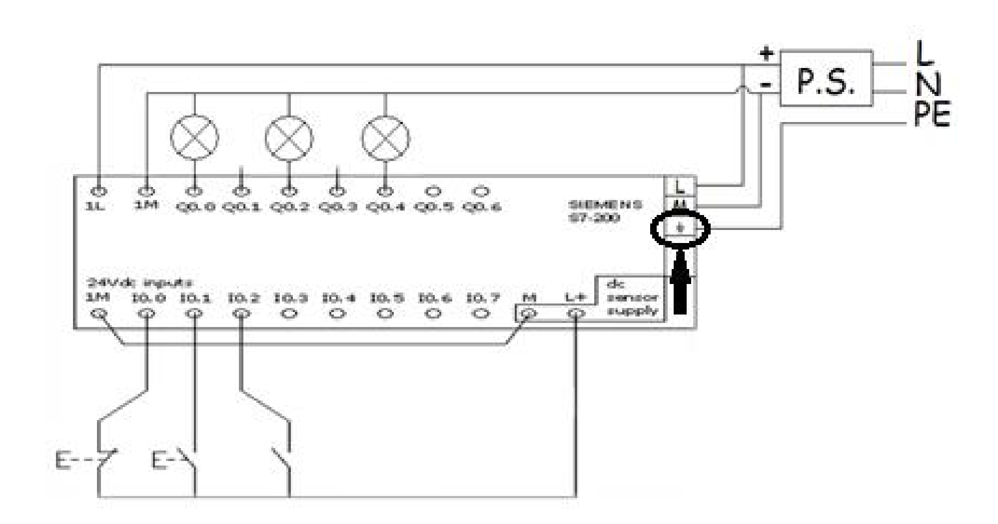
يستخدم نظام التوصيل بالأرضى لسببين: الأول لحماية الإنسان والثانى لحماية المعدة.

المبدأ الأئساسى لعمل التوصيل الأرضى هو أن قيمة مقومة الأرض أقل بكثير من قيمة أى جسم أخر حيث من المفروض أن لا تتعدى قيمة مقومة التوصيل الأرضى الـ ٢٥ أوم لأنه من المعروف أن التيار يسير فى الطريق ذات المقاومة الأقل لذالك فى حالة وجود أى فقدان للتيار فمن المؤكد أن التيار سيختار الانتقال إلى الأرض لأنها أقل من مقاومة الإنسان بكثير.

.



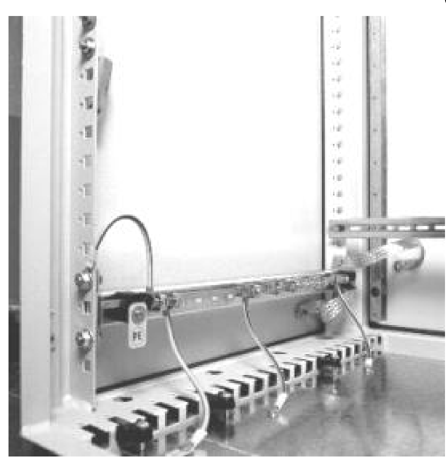
توصيل وحدة البرمجة بالأرضى:



ملاحظة:

يتم توصيل الطرف الأرضى على وحدة البرمجة لحمايتها من أى تسرب للتيار قد يؤدى إلى تلف الوحدة بالكامل, للمزيد عن التوصيل أنظر صفحة ٤٣ بالجزء الأول من كتاب برمجة التحكم المنطقى PLC.

توصيل الكبينة المعدنية بالأرضى:



ملاحظة:

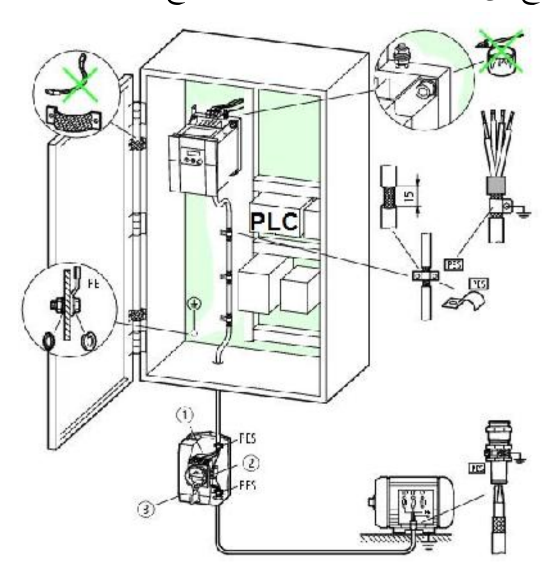
يتم توصيل الطرف الأرضى بجسم الكبينة المعدنية cabinet لحماية ما بداخلها من أجهزة حساسة بل ولحماية الإنسان الذى قد يلمس جسم الكبينة أثناء تسرب التيار الكهربى.

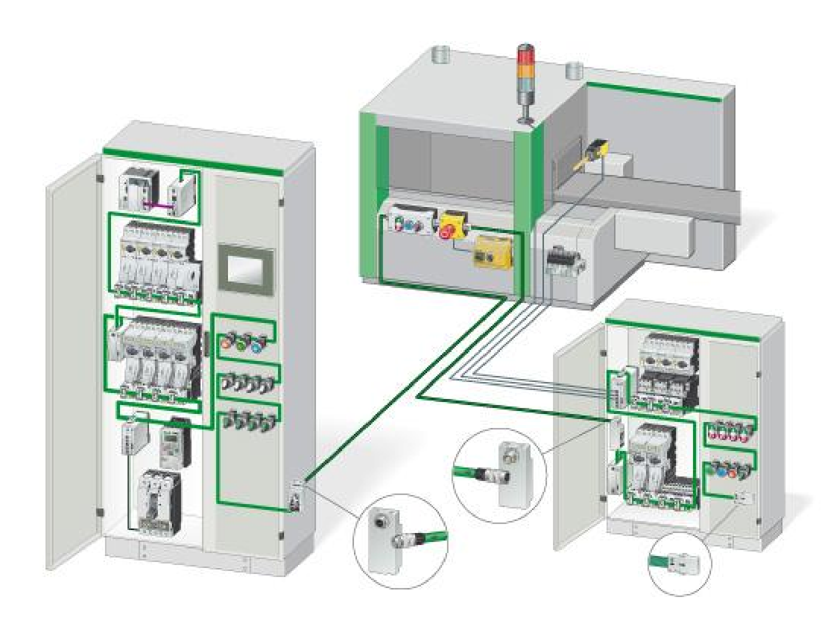
شكل التوصيل بالأرض:



ملاحظة:

بعد الانتهاء من تأريض كل المعدات والأجهزة يتم توصيل سلك التأريض بسيخ حديد يصل تحت الأرض بطول يتراوح من نصف متر فيما أكثر حسب نوع التربة المستخدمة.



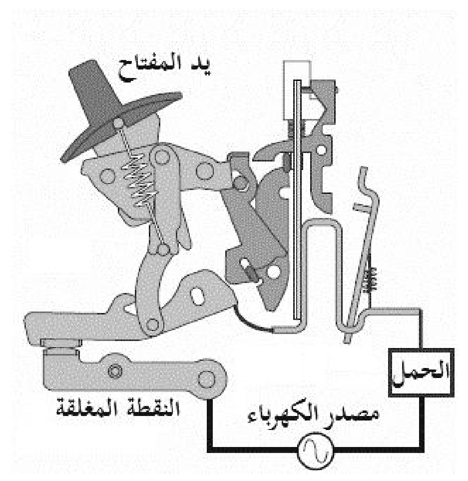


الحمايات

تعتبر الحمايات من العوامل الأساسية لضمان سلامة عمل وحدة البرمجة ولتجنب الأعطال بنسبة كبيرة ومن هذه الحمايات:

١- القاطع الكهربى "Circuit breaker"

تستخدم مفاتيح القاطع الكهربى الـ circuit breaker كحماية بين المصدر الرئيسى للكهرباء والأحمال المتصلة بالوحدة للحماية فى حاله أن قام أى من الأحمال بسحب قيمة عالية من التيار حيث يقوم القاطع الكهربى بفصل دائرة القوى التى تقوم بتغذية الأحمال, الشكل التالى يوضح تصميم المفتاح القاطع الكهربى.



شكل مفتاح القاطع الكهربى من الداخل



شكل مفتاح القاطع الكهربى من الخارج

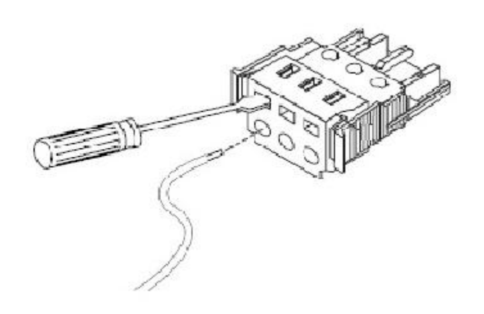
القاطع الحرارى "الأوفرلود – overload

تستخدم حماية الأوفرلود كحماية أساسية للأحمال ضد ارتفاع قيمة التيار حيث يتكون من معدنين مختلفين يتمدد الأول بطريقة مختلفة عن الأخر عند ارتفاع درجة الحرارة بسبب ارتفاع التيار المسحوب من الحمل, حيث يتم فصل جميع النقط الخاصة بأسلاك تغذية دائرة القوة "power" و دائرة التحكم "control"

ملاحظة:

- يعتمد تغيير نقاط الأوفرلود على تمدد القطع المعدنية عند مرور تيار أكثر من الحد المسموح ولذالك قد يتطلب هذا التمدد وقت كبير قد يصل إلى دقيقة تقريباً ولكن ليس من المفضل الانتظار كل هذا الزمن لذالك تستخدم المنصهرات الأوتوماتيكية لكى تفصل فى الحال.
- يتميز الأوفرلود بأنه يمكن أن يفصل حتى مع ارتفاع نسبة قليلة جداً للتيار بينما تحتاج المنصهرات إلى ارتفاع ملحوظ للتيار حتى تتم الاستجابة.





المفاتيح الأوتوماتيكية Automatic switch

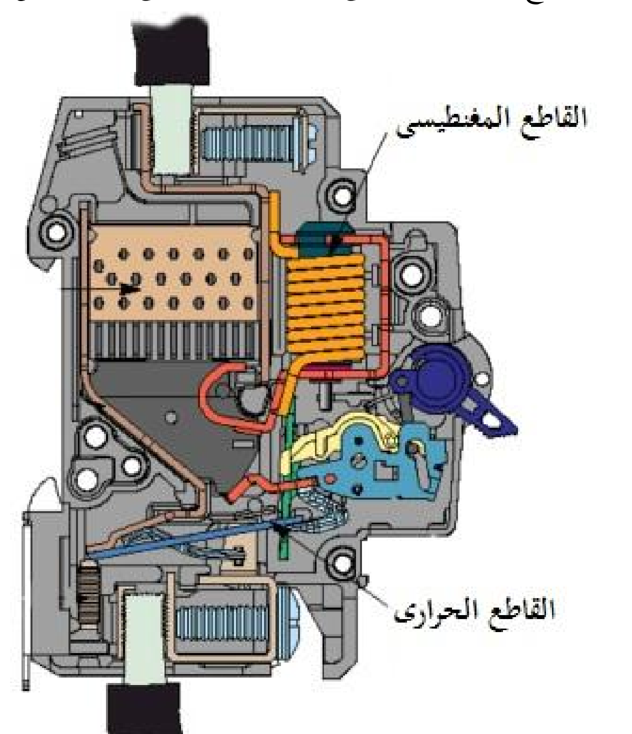
ينقسم المفتاح الأوتوماتيكى إلى قسمين:

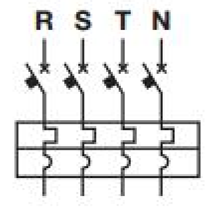
الأول – القاطع المغنطيسى: وهو القسم الذى يتمم عملية الفصل بواسطة المجال المغنطيسى الذائد الناتج عن زيادة فى التيار المسحوب من الحمل.

الأول – القاطع الحرارى: وهو القسم الذى يتمم عملية الفصل بواسطة التمدد الحرارى للشريحة المعدنية الناتج عن زيادة فى التيار المسحوب من الحمل.

ملاحظة:

- يتميز القاطع المغنطيسى فى سرعة الفصل فى حاله ارتفاع كبير فى التيار مقارنتاً بالقاطع الحرارى الذى قد يتطلب دقيقة كاملة لكى تتم عملية التمدد الحرارى.
- يتميذ القاطع الحرارى فى سرعة فى الفصل فى حاله ارتفاع ضئيل فى التيار مقارنتاً بالقاطع المغنطيسى الذى قد لا يفصل لأنه يتطلب ارتفاع كبير فى التيار لكى يتولد مجال مغنطيسى كافى للفصل.





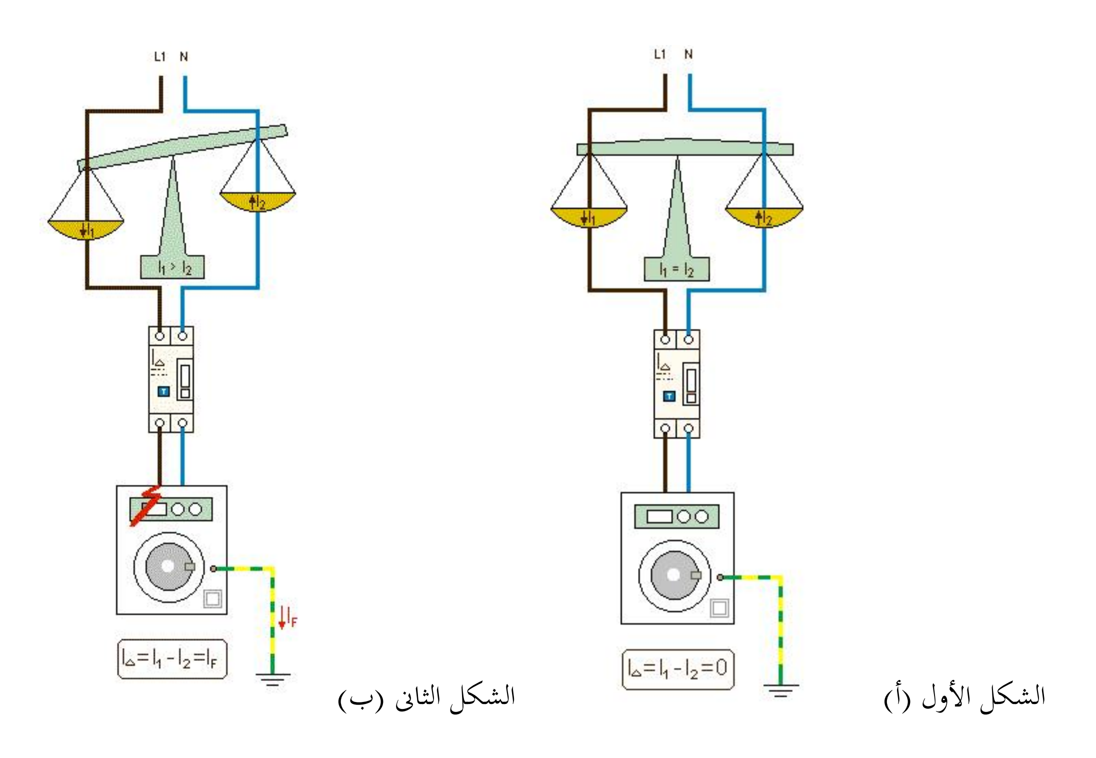
قيم خاصة بالمفاتيح الأوتوماتك

م	قيمة القدرة الكهربية للمفتاح [KVA]	قيمة التيار الطبيعى لعمل المفتاح [A]	أقصى قيمة للتيار يتحملها المفتاح فى حالة ألقفله [A]
١	50	72	1805
٢	100	144	3610
٣	160	230	5776
٤	200	288	7220
٥	250	360	9025
٦	315	455	11375
٧	400	589	14450
٨	500	722	18050
٩	630	910	22750
١٠	800	1156	28900
١١	1000	1444	36100

مفاتيح الحماية ضد التسريب Differential breaker

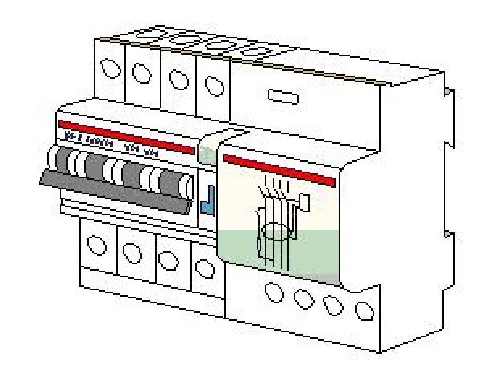
تعتمد طريقة عمل مفتاح الـ Differential على أن تكون قيمة التيار الداخل إلى الدائرة الكهربية يعادل نفس قيمة التيار الخارج من الدائرة الكهربية وذالك لأن التيار المار فى المفتاح (داخل إلى المفتاح) يقوم بتوليد مغنطيسى يتغير طردياً مع تغير قيمة التيار ولذالك فأن من الطبيعى أن يكون المجال المغنطيسى الناتج التيار الداخل يعادل نفس مقدار المجال المغنطيسى الناتج عن التيار الخارج ولذالك فأن عندما ينتج أى تغير بين ة التيار الداخل والخارج فتتغير تلقائياً قيمة المجال المغنطيسى الداخل والخارج ويحدث هذا فقط عندما يحدث أى تسريب للتيار فى جسم المعدة أو جسم الإنسان وتعتمد فكرة عمل المفتاح على التغير بين قيمة المجال المغنطيسى الناتج من التيار الداخل والمجال المغنطيسى الناتج من التيار الخارج.

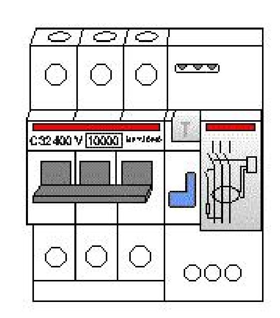
فى الشكل الأول (أ) قيمة التيار الداخل يعادل نفس قيمة التيار الخارج لأنه لا يوجد أى تسريب للتيار بينما فى الشكل الثانى (ب) قيمة التيار الداخل لا يعادل نفس قيمة التيار الخارج لأنه يوجد تسريب للتيار بجسم الغسالة كما هو موضح.

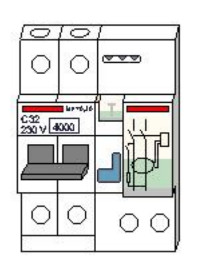


الشكل الثانى (ب)   الشكل الأول (أ)

الشكل الخارجى لمفتاح مانع التسرب







قيم خاصة بمفتاح الـ Differential.

م	قيمة التيار الطبيعى للمفتاح [mA]	أقصى قيمة للمقاومة الخاصة بالأرضى لـ 50V	أقصى قيمة للمقاومة الخاصة بالأرضى لـ 25V
١	10	5000	2500
٢	30	1660	830
٣	100	500	250
٤	300	170	85
٥	500	100	50
٦	1000	50	25

القانون المستخدم لحساب التيار الخاص بالمفتاح الـ differential هو التالى:

$$I = V/R$$

I: فرق التيار الكافى لتغيير حالة المفتاح.

V: قيمة الجهد الكهربى عند التلامس.

R: قيمة المقاومة الخاصة بالتأريض.

٢ - حمايات الفولت "Over-Under"

من الضرورى جداً التأكد بصفة مستمرة من قيمة الجهد سواء بالنسبة للوحدة, للمدخلات و للمخرجات. حيث أن تغير قيمة الجهد قد يؤدى إلى مشاكل عدة, فمثلاً:

- أرتفاع قيمة الجهد المتصل بوحدة البرمجة قد يؤدى إلى تلف الوحدة بالكامل.
- أرتفاع قيمة الجهد المتصل بالمدخلات قد يؤدى إلى تلف ببعض المدخلات أو بالوحدة بالكامل.
- ارتفاع قيمة الجهد المتصل بالمخرجات قد يؤدى إلى تلف ببعض المخرجات أو بالوحدة بالكامل.
- أنخفاض قيمة الجهد المتصل بالمدخلات قد لا يسمح بتغيير حاله المدخلات بالفعل.
- انخفاض قيمة الجهد المتصل بالمخرجات قد لا يسمح بتغيير حاله المخرجات بالفعل.

لذالك تستخدم الحمايات التالية كى تشير إلى أى حاله من ارتفاع أو انخفاض قيمة الجهد حيث يتم تشغيل ريليه فى حاله خروج قيمة الجهد عن الحد المسموح له.

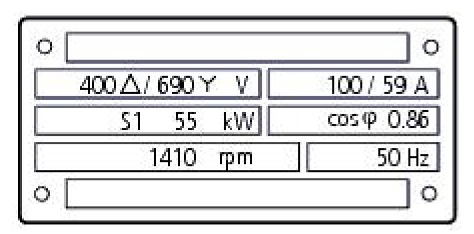


الرموز الخاصة بالحمايات

م	أسم	رسم
١	حماية ضد القفلة الكهربية	
٢	حماية ضد ألقفله الكهربية وارتفاع التيار	
٣	حماية ضد ألقفله الكهربية وارتفاع التيار و تسرب التيار	
٤	حماية ضد تسرب التيار	
٥	حماية ضد ألقفله الكهربية وارتفاع التيار مع منصهر	

٣ – ريليهات الحماية "Interface relay"

تستخدم الريليهات كحماية بين وحدات البرمجة والأحمال المتصلة بالوحدة لحمايتها فى حاله أن قام أى من الأحمال بسحب قيمة عالية من التيار حيث يتم تركيب ريليهات حسب اللوحة المثبتة على الحمل كما بالشكل التالى



فمثلاً حسب لوحة البيانات السابقة يتم شراء ريليه بحيث تتحمل نقطة الريليه قيمة التيار المسحوبة من الحمل.

ملاحظة:

- من المعروف أن النقطة المغلقة للريليه تتحمل تيار أقل من قيمة التيار الذى تتحملة النقطة المفتوحة لنفس الريليه حيث أن النقطة المغلقة تبقى متصلة فترة أكبر من تلك المفتوحة.
- تنقسم الريليهات إلى نوعين: النوع الميكانيكى و النوع الألكترونى ويتميز الريليه الألكترونى بسرعة الاستجابة حيث أنة لا يحتوى على أى حركة ميكانيكية.

– الريليه الميكانيكى.

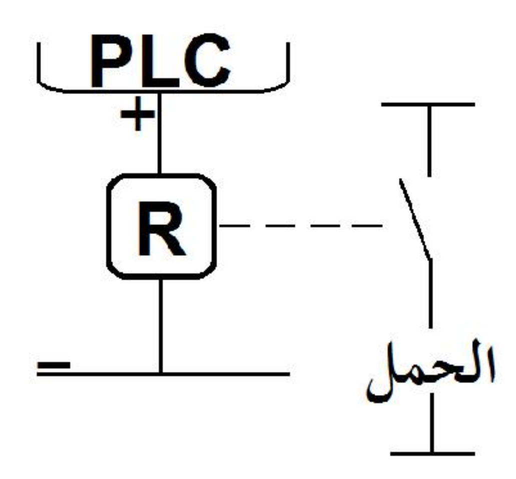


– الريليه الإلكترونى.



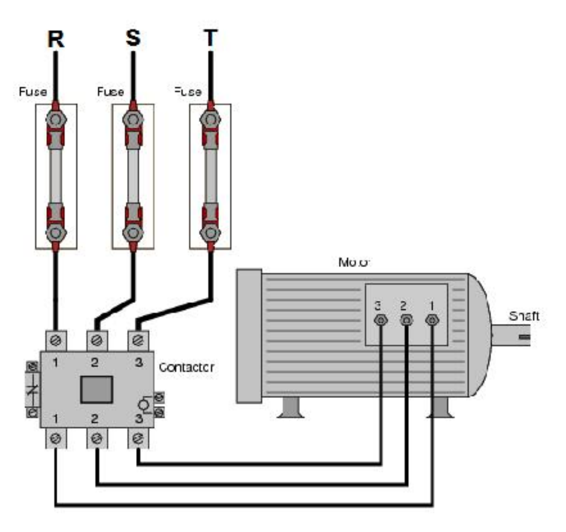
كيفية توصيل الريليه:

يتم توصيل الريليه الألكترونى أو الميكانيكى بوحدة البرمجة PLC ومن ثم يتم تغذية النقاط الخاصة بالريليه لتشغيل الأحمال بمختلف قدرتها ويراعى فى ذالك أن تتفق مواصفات نقاط الريليه مع مواصفات الحمل.



الكونتكتور

يعمل الكونتكتور بنفس طريقة عمل الريليه الميكانيكى ولكن مع العلم بأنه يمكن للكونتكتور أن يعمل بقدرات أعلى بكثير من تلك الخاصة بالريليهات ولذالك يتم استخدام الكونتكتور مع المحركات ذات الأوجه الثلاثة كما بالشكل التالى وليس الريليه.



٤ – المنصهرات "Fuses"

تستخدم المنصهرات كحماية بين مصدر الكهرباء و وحدات البرمجة وبين مصدر الكهرباء والأحمال المتصلة بالوحدة للحماية من ارتفاع التيار يتم تركيب المنصهرات حسب اللوحة المثبتة على الحمل حيث يعتمد على قيمة التيار الخاص بالأحمال المتصلة, كما بالشكل التالى



٥ – البطاريات "Batteries"

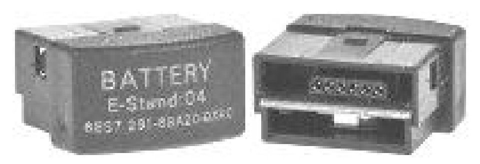
تستخدم أيضاً البطاريات كحميات ذات أهمية فائقة فى حاله انقطاع الكهرباء وخصتاً بالنسبة للصانعات الثقيلة أو فى الصناعات التى لا يمكن أن تتوقف قط, حيث أن انقطاع الكهرباء لمجرد لحظات قد يسبب كارثة لسببين:

السبب الأول أن كان قد تم تسجيل البرنامج على ذاكرة من النوع المتطاير ففى حالة انقطاع التيار وبالأخص أن كانت لا توجد بطارية متصلة بوحدة البرمجة فسوف يؤدى هذا إلى فقدان البرنامج.

السبب الثانى هو أن فى بعض خطوط الإنتاج قد تحدث العديد من المشاكل فى حال انقطاع الكهرباء وفى حالة عدم وجود بطارية أيضاً حيث قد تتغير بعض القيم المتعلقة بالمؤقتات الزمنية والمتعلقة بالمتغيرات أيضاً.

لذالك من الضرورى جداً وجود مصدر بديل للكهرباء والمصادر البديلة للكهرباء قد تكون بطارية مختصة للحفاظ على البرنامج أو بطارية لاستمرار عمل وحدة البرمجة دون توقف حتى عند أنقطاع مصدر الكهرباء.

النوع الأول: Battery



بطارية تتغير حسب نوع وحدة البرمجة

النوع الأول: UPS



بطارية خارجية تعمل كبديل فى حالة انقطاع مصدر الكهرباء

تنقسم بطاريات الـ UPS إلى نوعين:

الأول: Online Uninterruptable Power Supply

حيث تتميز البطاريات الـ online بالعمل بطريقة مستمرة سواء كان يوجد مصدر للكهرباء أو لا يوجد ويستخدم هذا النوع من البطاريات مع الأحمال التى لا يمكن أن تتوقف حتى لو للحظة واحدة مثل أجهزة الـ PLC, أجهزة الكمبيوتر, شبكات المحمول, المستشفيات, الخ.

فى حالة وجود مصدر للطاقة (الحكومة):

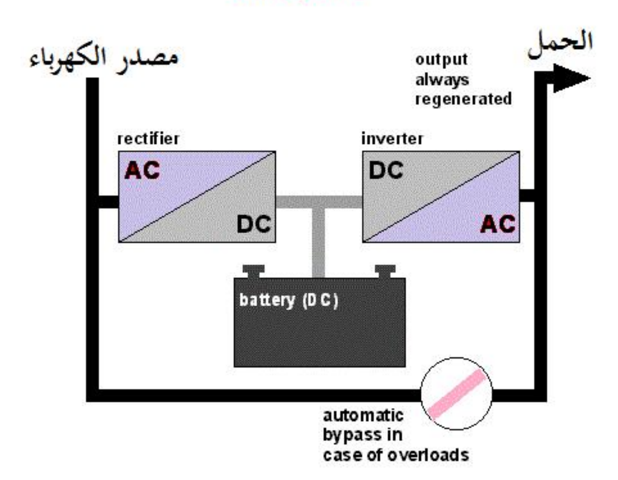
تقوم كهربة الحكومة بشحن البطاريات بصفة مستمرة وتقوم البطاريات بتغذية جميع الأحمال.

فى حالة وجود مصدر بديل للطاقة (المولدات الكهربية):

تقوم كهربة المولدات بشحن البطاريات بصفة مستمرة وتقوم البطاريات بتغذية جميع الأحمال.

ملاحظة:

- الوقت بين انقطاع الكهرباء وتوفير أى مصدر بديل كالمولد يجب أن لا يتعدى الدقيقة "حوالى ٣٠ ثانية" وخلال هذا الوقت تكون البطاريات هى المسئول الأول والأخير عن توفير الطاقة.
- من المفروض أن تكون البطاريات كافية لتوفير الطاقة لمدة لا تقل عن ٢٠ دقيقة تقريباً لضمان عمل المصدر البديل "المولد الكهربى"



الثانى: Offline Uninterruptable Power Supply

حيث تعمل البطاريات الـ offline و تسمى أيضاً بالبطاريات الأستعدادية "standby UPS"بطريقة الـ bypass "التجنب" يستخدم هذا النوع من البطاريات مع الأحمال الأقل أهمية حيث يمكن للمصدر الطاقة أن ينقطع عنها للحظات معدودة حيث تبدأ هذه البطاريات فى العمل بعد تغيب مصدر الكهرباء الرئيسى فتستخدم مع الأحمال التى من الممكن أن تتوقف للحظات مثل الإضاءة, كاميرات المراقبة, الخ.

فى حالة وجود مصدر للطاقة (الحكومة):

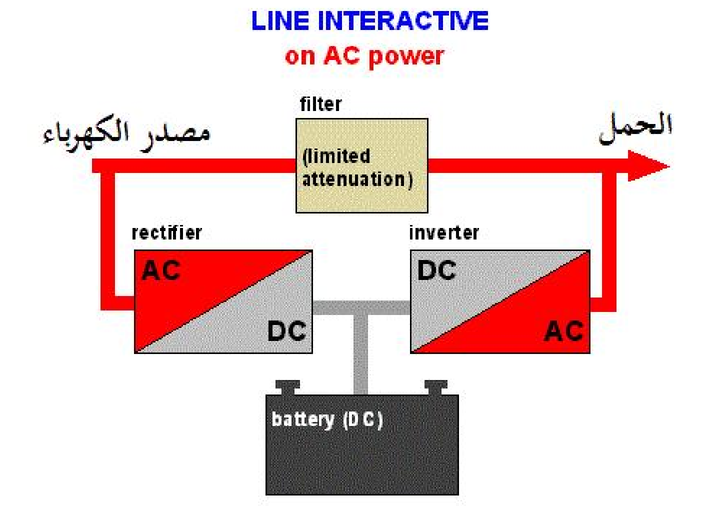
تقوم كهربة الحكومة بشحن البطاريات بصفة مستمرة وتقوم كهرباء الحكومة أيضاً بتغذية جميع الأحمال بطريقة مباشرة دون استخدام البطاريات وتسمى هذه الطريقة بنظام التجنب "bypass system".

فى حالة عدم وجود مصدر للطاقة (انقطاع الكهربية):

تقوم البطاريات بتغذية جميع الأحمال حتى يعود المصدر الرئيسى للكهرباء .

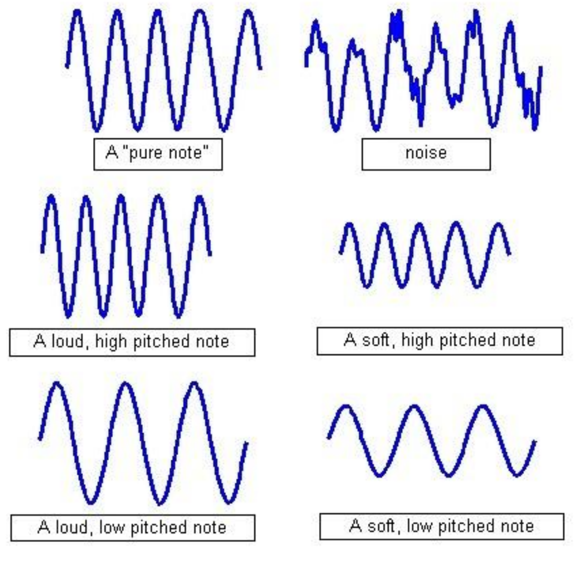
ملاحظة:

يستخدم هذا النوع مع الأحمال الأقل أهمية كما سبق وذكرنا.



ملاحظة:

تحتوي بطاريات الـ UPS على منظمات للجهد لضمان عدم وجود تشوهات بالموجة.



الشكل العام لموجات التيار الـ AC

مفتاح التحويل الأوتوماتيكى الـ ATS

يستخدم مفتاح التحويل الأوتوماتيكى الـ ATS وهو اختصار لكلمة " Automatic Transfer Switch"

للتحويل من مصدر طاقة إساسى مثل كهربة الحكومة إلى مصدر طاقة بديل مثل المولدات.



مفتاح التحويل الأوتوماتيكى.

ملاحظة:

- تحتوي دائرة تحكم الـ ATS على مفتاح اختيار Selector Switch أربعة أوضاع:

الوضع الأول: اختيارOFF.

الوضع الثاني: اختيار شركة الكهرباءMains Supply.

الوضع الثالث: اختيار المولد Generator.

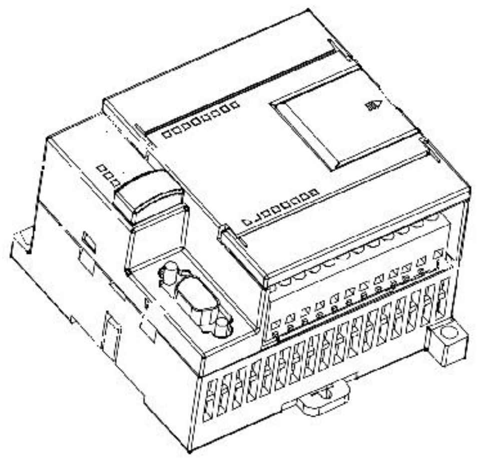
الوضع الرابع: اختيار أوتوماتيكىAuto.

- يمكن تنفيذ دائر كهربية تحل محل مفتاح التحويل الأوتوماتيكى باستخدام كونتاكتورات مؤقتات زمنية.



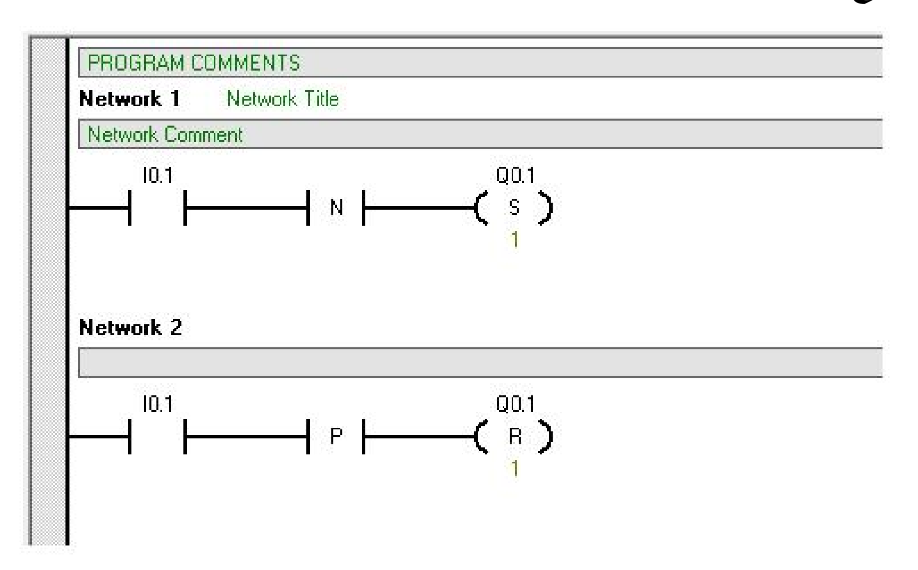
دائرة كهربية تعمل بنفس فكرة مفتاح الـ ATS.

يمكن تنفيذ دائرة منطقية تحل محل مفتاح التحويل الأوتوماتيكى باستخدام وحدة برمجة صغيرة كما بالبرنامج التالى.



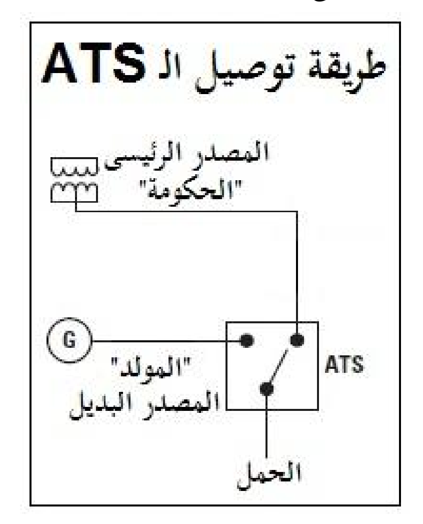
وحدة برمجة منطقية PLC.

البرنامج



بحيث تمثل النقطة I0.1 نقطة مفتوحة من ريلية يعمل بواسطة مصدر الكهرباء الأساسى "الحكومة" فعند انقطاع تلك المصدر تفتح نقطة المفتاح I0.1 فيعمل الخرج Q0.1 الذى يقوم بتشغيل المصدر البديل للطاقة "المولد الكهربى" إلى أن يعود المصدر الرئيسى مرة أخرى فتغلق نقطة الريليه I0.1 فيتوقف الخرج Q0.1 وهكذا.

طريقة توصيل مفتاح التحويل الأوتوماتيكى:

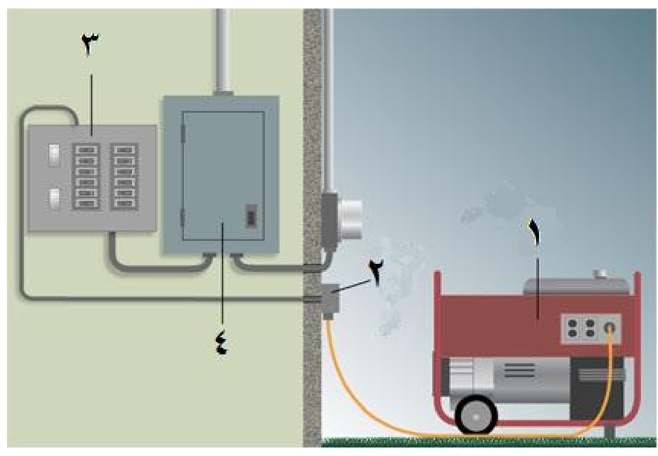


١- المولد الكهربى (مصدر الطاقة البديلة).

٢- بواط (مخرج الكهرباء من الحائط).

٣- مفتاح التحويل الأوتوماتيكى "ATS".

٤- كهربة الحكومة (المصدر الرئيسى للطاقة).



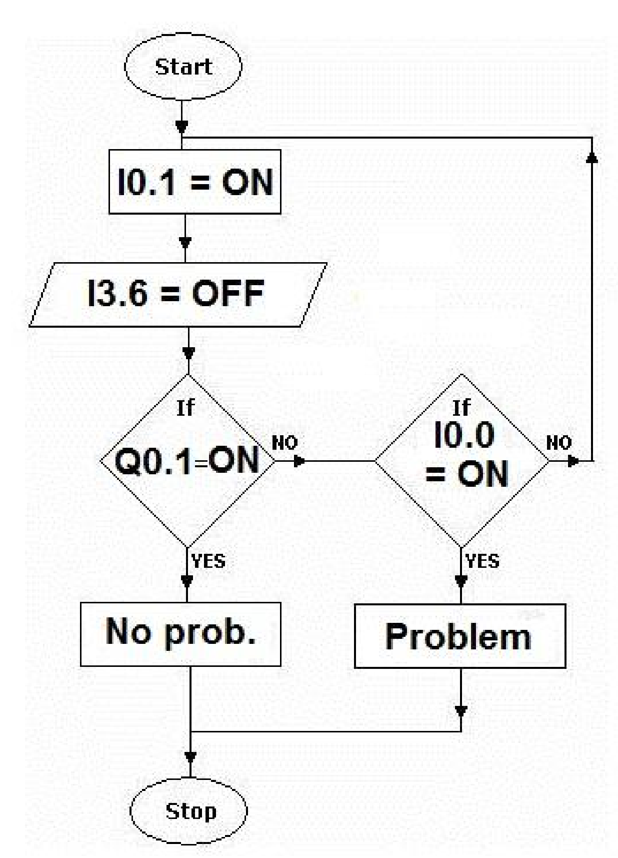
صورة توضيحية لتوصيل مفتاح الـ ATS.

من الأشياء الضرورية الأخرى لسرعة اكتشاف وحل الأعطال هى الكاتالوجات والمنيول قد تتوفر مع المعدات الكبيرة بعض المنيول الخاصة باختبار المكينة "testing manual" حيث يستخدم لاختبار عمل المكينة فمثلاً:

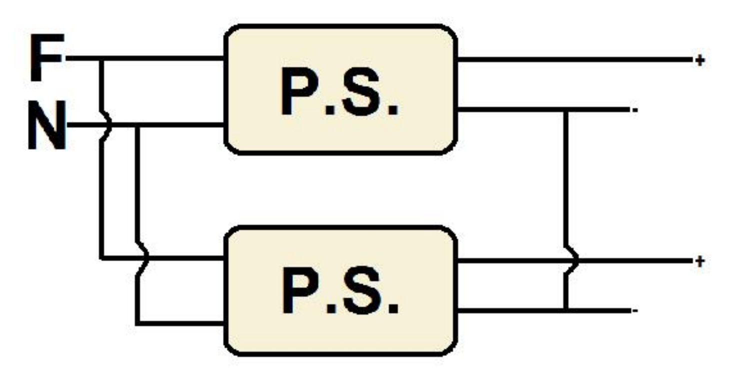
أولاً: أضغط على المفتاح I0.1
ثانياً: أفتح مفتاح نهاية المشوار I3.6
ثالثاً: أغلق مفتاح الإيقاف I0.0

يجب أن يعمل المحرك لليمين فى ما عدا ذالك فهذا يدل على وجود مشكلة.

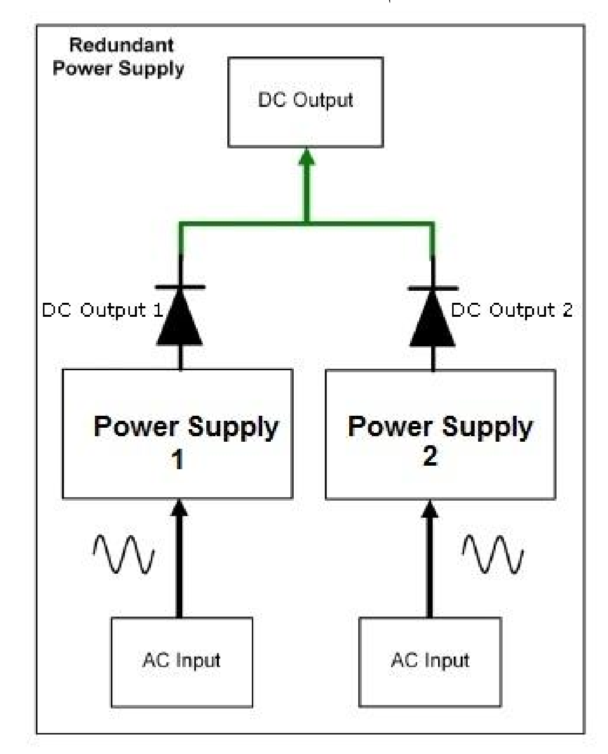
يمكن أن يتم شرح نفس الخطوات السابقة ولكن باستخدام مخطط flow chart كالتالى:



- فى حالة توصيل أكثر من مصدر للطاقة المستمرة power supply على نفس الوحدة يفضل توصيل الطرف السالب للاثنين معاً لضمان تساوى القيمتين balance.

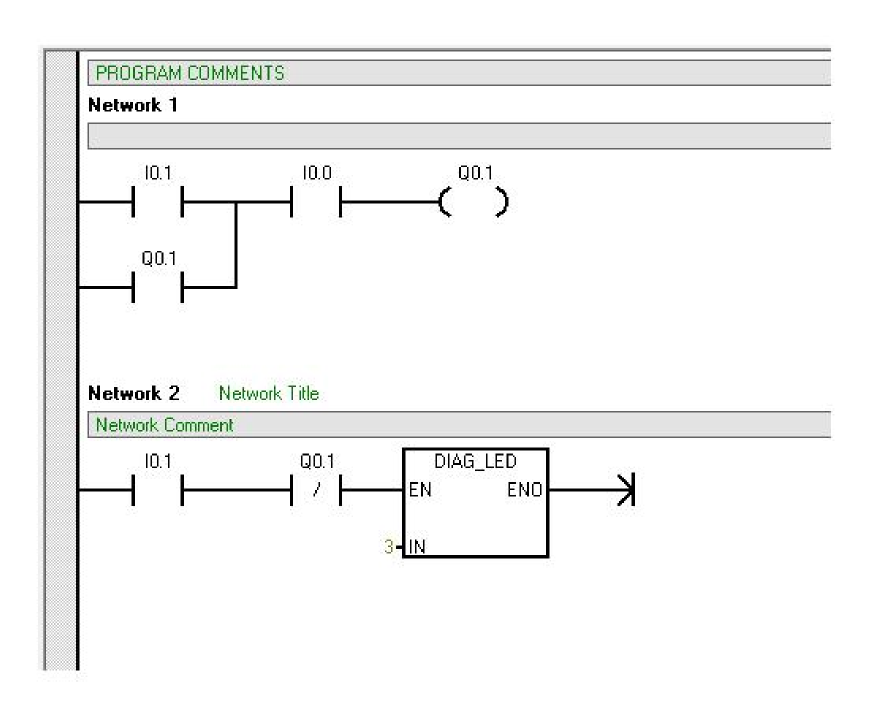


- مع وحدات البرمجة التى تقوم بأعمال أساسية وهامة جداً يتم استخدام نظام يسمى بال redundant system حيث يقوم على الانتقال من مصدر للطاقة إلى مصدر أخر بديل دون أن تتوقف وحدة البرمجة ويتوفر هذا النظام فى وحدات البرمجة S7-400.



لمبة التشخيص Diagnostic LED

يمكن استخدام لمبة التشخيص للإضاءة فى حالات خاصة يمكن تحديدها بواسطة البرنامج, فمثلاً فى البرنامج التالى قد تم برمجة اللمبة بحيث تضاء فى حاله عدم تشغيل المحرك عند الضغط على مفتاح البدء.



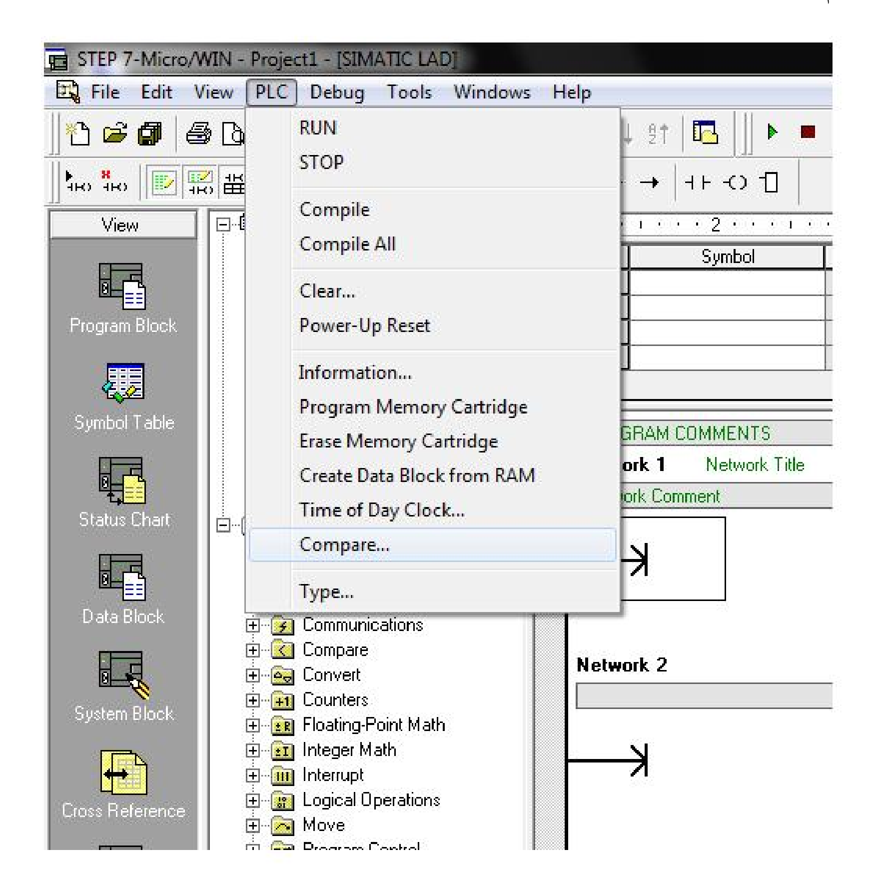
ملاحظة:

يفضل عدم استخدام هذه اللمبه لعدم التخبط بين الاستخدام الأساسى لها والاستخدام الأخر كما فى البرنامج السابق.

للمقارنة بين البرنامج الموجود على الكمبيوتر والبرنامج الموجود على وحدة البرمجة للتأكد من أنه نفس البرنامج.

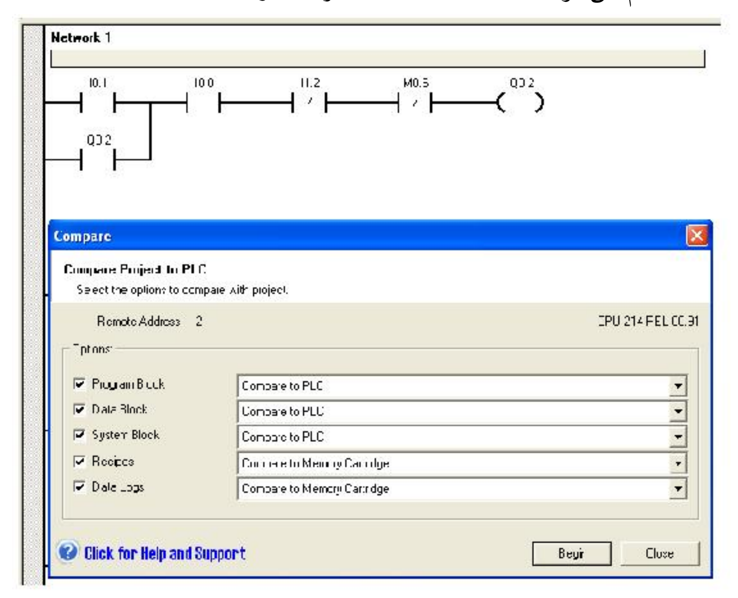
أولاً.

يتم اختيار compare من قائمة الـ PLC.



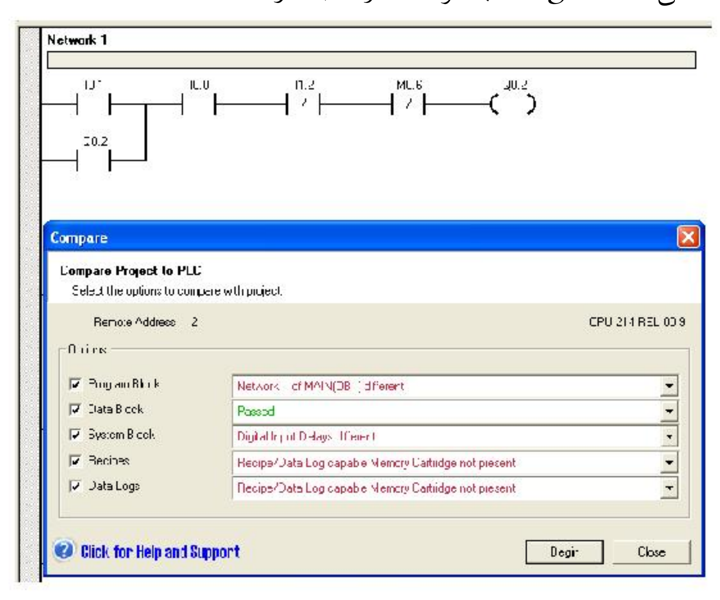
ثانياً.

تظهر صفحة يتم من بواسطتها تحديد الأشياء المراد مقارنتها.



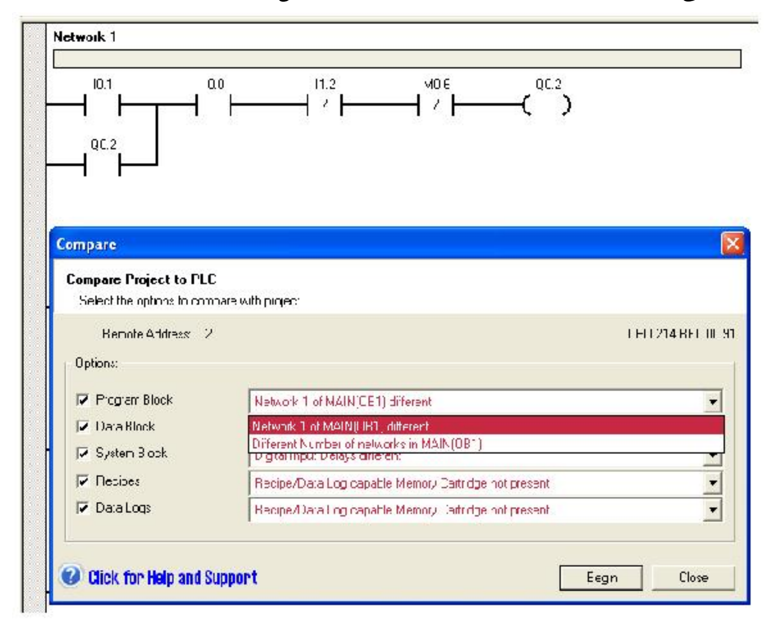
ثالثاً.

بالضغط على كلمة Begin تبدأ وحدة البرمجة بالمقارنة.



رابعاً.

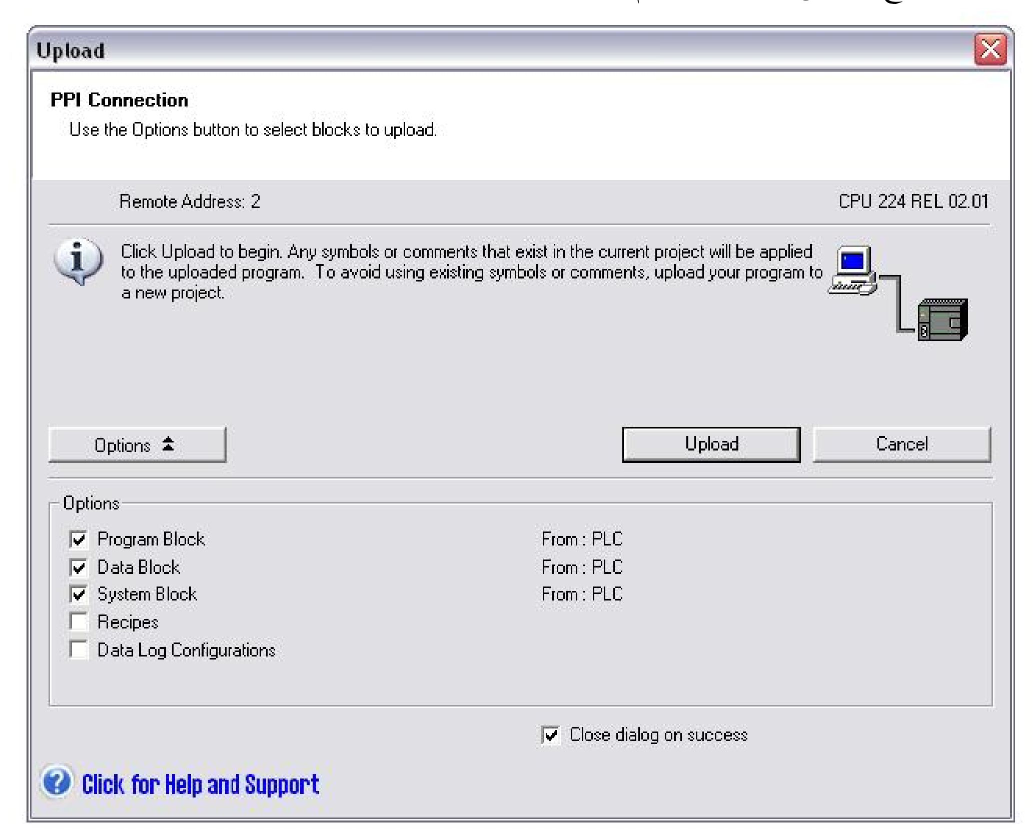
بالضغط على الرسالة الظاهرة تظهر قائمة بأكثر تفاصيل.



ملاحظة:

يعتبر أمر المقارنة أساسى فى حالة التأكد من أن البرنامج الذى على وحدة البرمجة هو نفس البرنامج الموجود على جهاز البرمجة "الكمبيوتر" خصتاً بأن فى بعض أجهزة البرمجة يتم وضع كلمات المرور لعدم رؤية البرنامج الموجود على وحدة البرمجة.

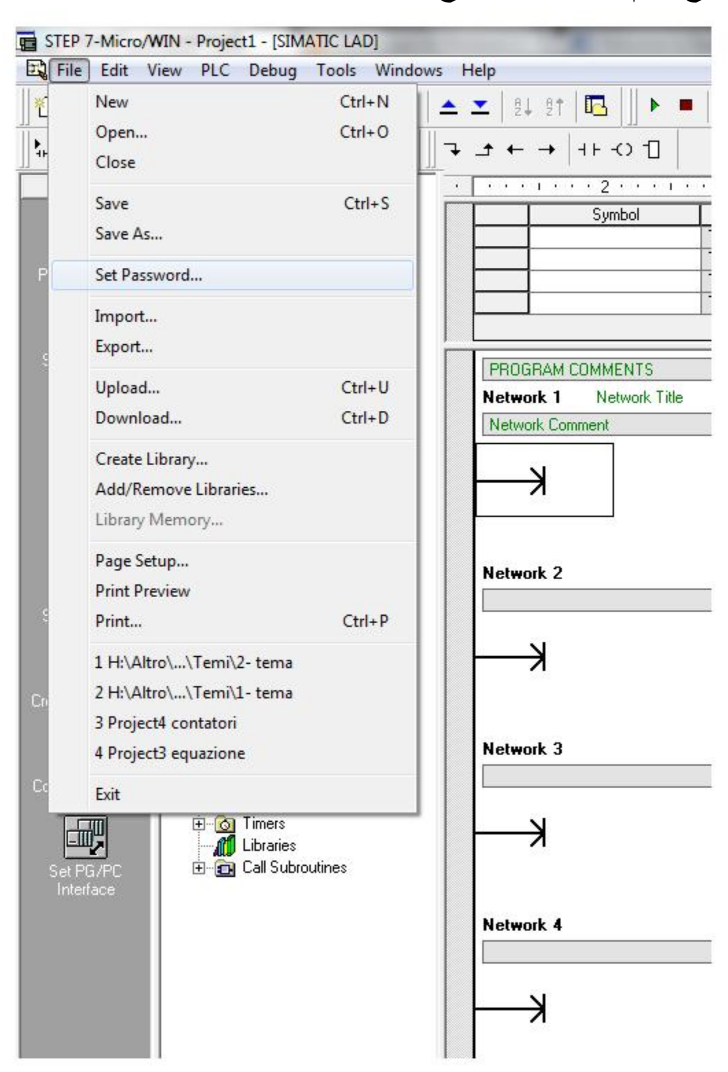
لسحب البرنامج من على وحدة البرمجة يتم تطبيق أمر Upload:

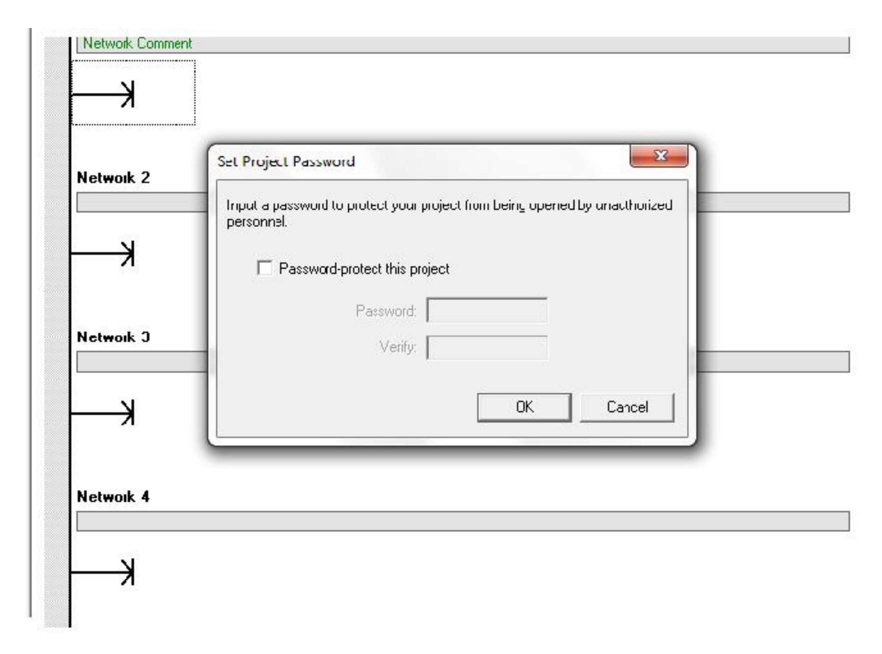


ملاحظة:

يمكن سحب البرنامج من على وحدة البرمجة دون مشاكل إلا في حالة وجود كلمة مرور تمنع هذا الأمر.

طريقة حفظ البرنامج برقم سرى حتى لا يفتح من قبل أى شخص





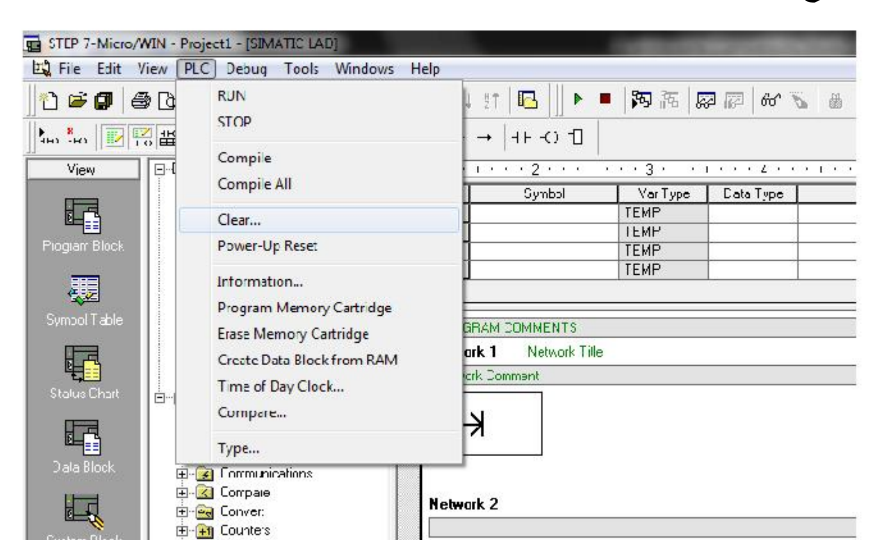
عند فتح الملف مرة أخرى سوف تظهر الرسالة التالية حيث يطلب منك كلمة المرور لإكمال عملية الفتح.



ملاحظة:

يعتبر هذا النوع من كلمات المرور من كلمات المرور الأقل قوة حيث تمنع فتح الملف لكن في الواقع يمكن بكل سهولة سحب البرنامج من على وحدة البرمجة.

لمسح البرنامج فى حالة وجود PASSWORD

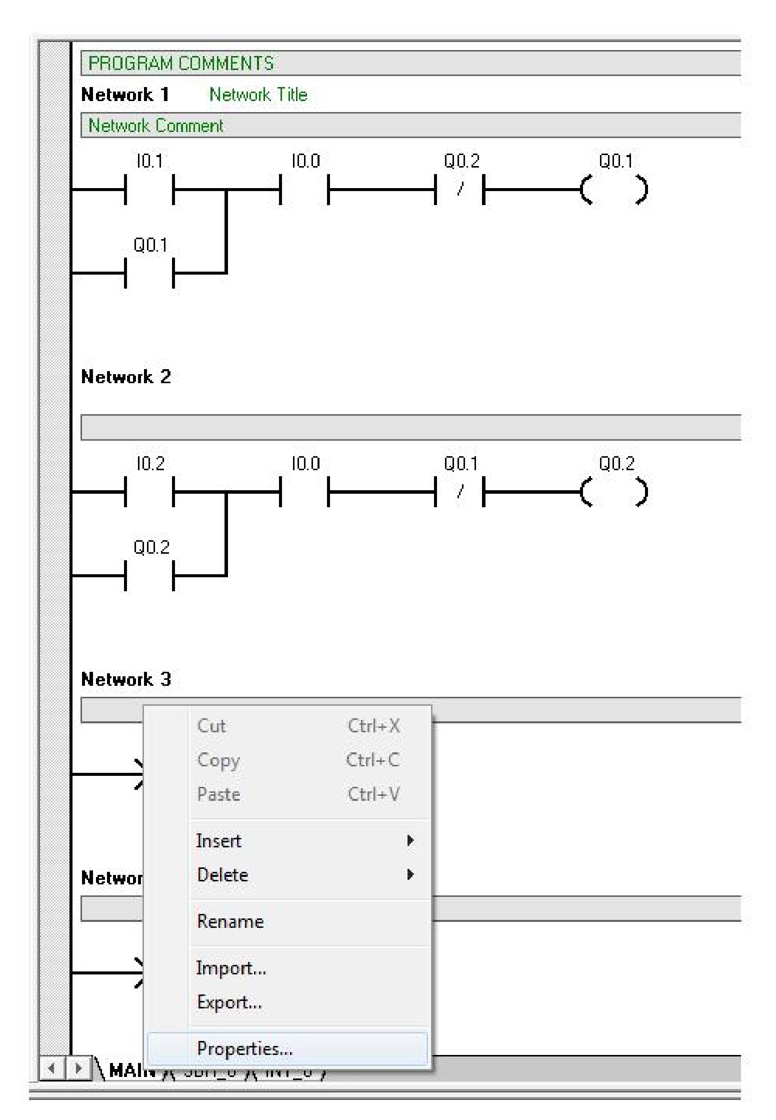


ملاحظة:

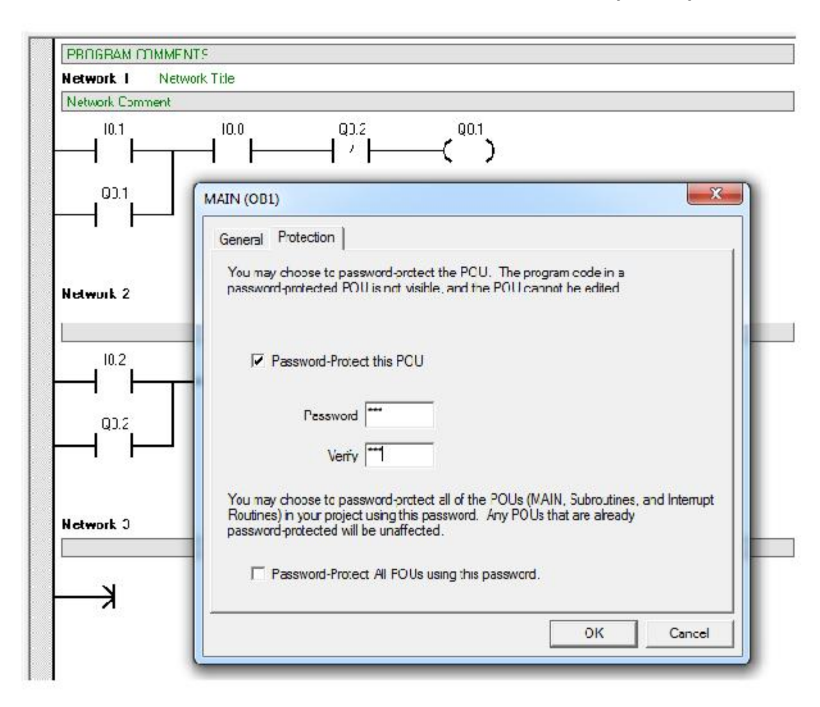
- لمسح كلمة المرور **والبرنامج** أيضاً يستخدم أمر clear حيث تصبح وحدة البرمجة جاهزة من جديد لاستقبال أى برنامج.
- حل أخر بحيث تكتب كلمة CLEARPLC بدلاً من كلمة المرور بحيث سيتم مسح كل شيء.

توجد طريقة أخرى وهى المفضلة لتطبيق كلمة المرور لحفظ البرنامج.

أولاً. يتم الضغط على الخصائص.



ثانياً. يتم كتابة كلمة المرور المطلوبة.



ثالثاً. بالضغط على OK يختفى البرنامج بالكامل لكنة يبقى موجود كما بالشكل التالى.



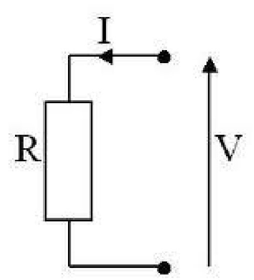
الباب السادس

ملاحظات هامة

- شرح فـــــــرق الجهـــــــــــــــــد.
- حساب سمك الســــــــــــــــــــلك.
- أنواع وألوان الأســـــــــــلاك المستخدمة.
- أعطــــــــال جهاز الـ Power Supply.
- أعطال الريليـــــــــــــــــهات الخارجية.
- تنفيذ دائــــــــــــــــــــــــرة الإلرم.
- مفـــــــــك التستر ولوحـــــات التثبيت.
- التلامس المباشر وغير المباشر للكهربـــاء.
- السرينة والجرس الكهربى للإعطـــــــــال.
- مفتـــــــــــــــــــــــــاح الطوارىء.
- بعض الكاتولجات لتوصيل وحدة الـ PLC.

قيمة فرق الجهد

تنقسم قيم الجهد إلى عدة قيم مختلفة وتعتمد على النقط التى سيتم التوصيل بينها:
فمثلاً بقياس قيمة الجهد بين طرف الـ N و أى طرف أخر من الفاز فتكون قيمة الجهد هى ٢٢٠ فولت, أما عند قياس الجهد بين أى طرف من الفاز وأى طرف أخر من الفاز فتكون قيمة الجهد هى ٣٨٠ فولت.
فى بعض الأحيان يتم استخدم جهد غير متردد لتغزيه وحدة البرمجة الـ PLC ويتم هذا باستخدام محول كهربى خافض للجهد من ٢٢٠ إلى ٢٤ فولت ومن ثم يتم توحيد التيار إلى تيار مستمر بدلاً من تيار متردد.

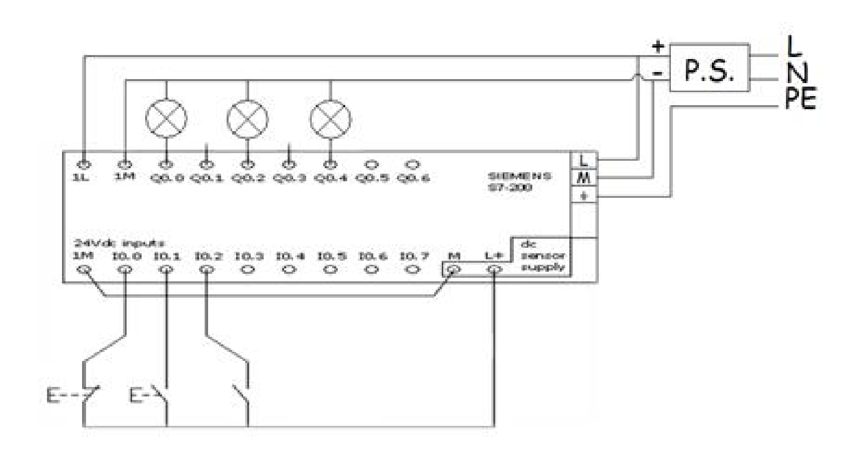


الحسابات الخاصة بالأسلاك

تعتبر الحسابات الخاصة بسمك السلك من الأشياء التى من الضرورى أخذها فى الاعتبار قبل البدء فى التنفيذ العملى و تنقسم الحسابات إلى جزئين:

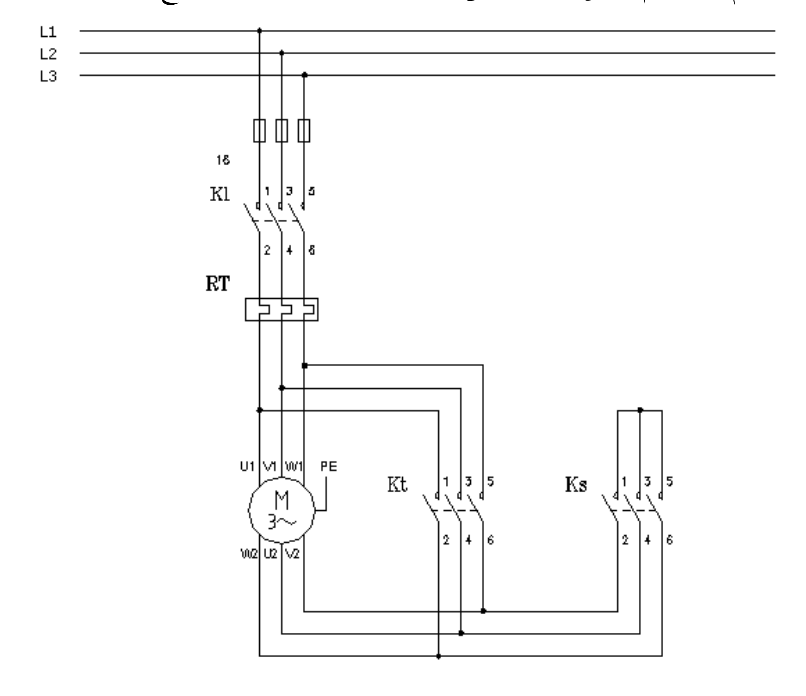
الأول – الخاص بالأسلاك الموصلة بوحدة البرمجة PLC

ولكن فى حقيقة الأمر لا يكون هناك احتياج لعمل أى حسابات خاصة بهذه الأسلاك نظراً لقلة التيار الذى يمر بها فلذالك يتم استخدام سلك بمساحة مقطع تساوى ١مم فقط دون الحاجة إلى أى حسابات ويتم عمل نفس الشىء بالنسبة للأسلاك الموصلة بالخرج لأن فى حقيقة الأمر قيمة التيار الذى يمر فى الأسلاك المثبتة بمخرج وحدة البرمجة هو قليل جداً قد لا يتعدى الـ ٣٠٠ ملى أمبير وهو ناتج عن التيار المسحوب من تغذية وحدة البرمجة للريليهات الميكانيكية المثبتة كحماية بين وحدة البرمجة والحمل.



الثانى – الخاص بالأسلاك الموصلة بدائرة القوة

فى هذه الحالة يكون من الضرورى جداً عمل الحسابات الخاصة بهذه الأسلاك نظراً لقيمة التيار التى قد تكون مرتفعة وتتطلب أسلاك ذات مساحة مقطع أكبر لكى تتحمل قيمة التيار المسحوبة من قبل الحمل المتصل بالريليهات فلذالك يتم استخدام بعض القوانين التى تساعد فى تحديد مساحة مقطع السلك.



حسابات سمك السلك

تعتبر الحسابات الخاصة بسمك السلك من الأشياء التى من الضرورى أخذها فى الاعتبار قبل البدء فى التنفيذ العملى.

القوانين

يتم حساب القدرة الفعالة للمحرك باستخدام القانون التالى بحيث توجد جميع القيم المطلوبة على اللوحة المثبتة بالمحرك.

$$P = \sqrt{3} . V . I . \cos\alpha$$

يتم حساب القدرة غير الفعالة للمحرك باستخدام القانون التالى بحيث توجد جميع القيم المطلوبة على اللوحة المثبتة بالمحرك.

$$Q = \sqrt{3} . V . I . \sin\alpha$$

يتم حساب القدرة المركبة للمحرك باستخدام القانون التالى, قانون فيثاغورث.

$$S = \sqrt{(P^2 + Q^2)}$$

يتم حساب التيار الكلى للمحرك باستخدام القانون التالى.

$$I = \frac{S}{\sqrt{3}\,.V}$$

يعوض عن الرمز التالى برقم ثابت وهو يشير إلى قيمة التيار التى يمكن أن يتحملها سلك بمقطع ١مم.

$$\varepsilon \approx 4$$

يتم حساب مقطع السلك الملائم لتلك الدائرة باستخدام القانون التالى.

$$Sez = \frac{I}{\varepsilon}$$

يتم حساب القدرة الفعالة للمحرك بالاعتماد على معامل الكفاءة باستخدام القانون التالى.

$$P_a = \frac{P_r}{\tau}$$

يتم حساب القدرة غير الفعالة للمحرك بالاعتماد على معامل القدرة باستخدام القانون التالى

$$Q = P_a \,. \tan\varphi$$

مثال لحساب مقطع السلك

المعطيات

قيمة الجهد ٣٨٠ فولت.

قيمة معامل الكفاءة ٠,٧٢

قيمة معامل القدرة ٨,٠

$$P_r = 4\ [HP]$$

$$\tau = 0{,}72$$

$$\cos\varphi = 0{,}8$$

المطلوب

معرفة مقطع السلك المطلوب استخدامه

الحساب

$$P_r = 4\ .736 = 2944\ [W]$$

$$P_a = \frac{P_r}{\tau}$$

$$P_a = \frac{2944}{0{,}72} = 4088{,}88\ [W]$$

$$Q = P_a\ .\tan\varphi$$

$$Q = 4088{,}88\ .0.75 = 3066{,}66\ \ [VAR]$$

$$S = \sqrt{(4088{,}88)^2 + (3066{,}66^2)}$$

$$S = 5111{,}1\ [VA]$$

$$I = \frac{5111,1}{\sqrt{3}\,.380}$$

$$I = 7,765\ [A]$$

$$\varepsilon \approx 4$$

$$Sez = \frac{7,765}{4}$$

$$Sez = 1,94\ \approx 2,5\ [mm^2]$$

ملاحظة:

عادتاً لا يتم شراء السلك بنفس قيمة المقطع المحسوبة باستخدام القوانين بل يتم شراء المقطع الأكبر والمتاح فى السوق لضمان عدم تلف السلك عند أى أرتفاع طفيف للتيار.

أنواع الأسلاك

من الأنواع الأكثر استخداماً للسلك الكهربى السلك المسمط و السلك الشعر ولكن يفضل استخدام السلك الشعر وهذا للميزات التالية:

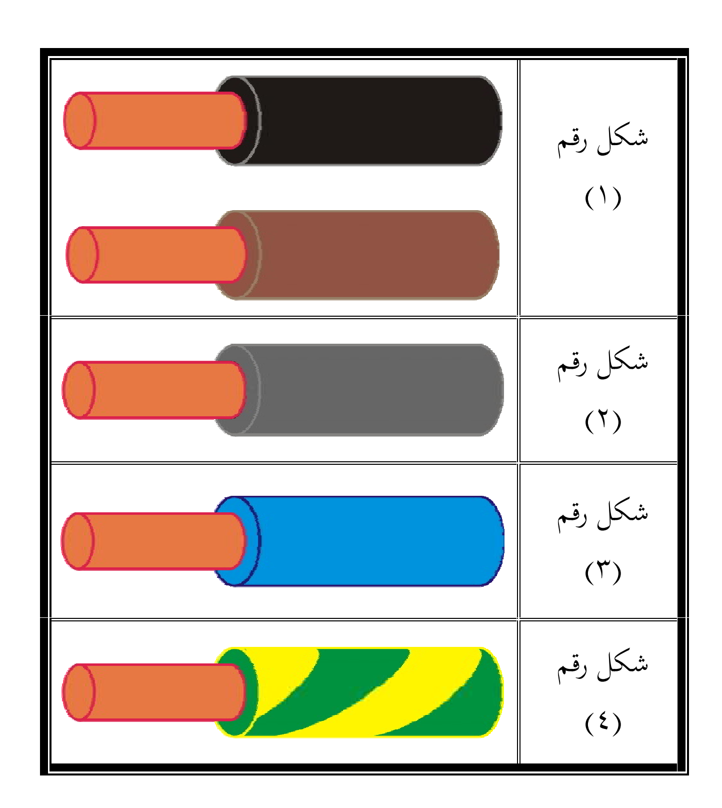
- يبرد السلك بطريقة أفضل بسبب التلامس الدائم مع الهواء.
- يتحمل قيمة أكثر من التيار الكهربى بالمقارنة بالسلك المسمط.
- يتحمل الكثير من التنى والتشكيل أثناء العمل.



ألوان الأسلاك

من الأشياء الأساسية جداً فى مبادئ الكهرباء بطريقة عامة وفى مجال استخدام وحدات البرمجة بطريقة خاصة ألوان الأسلاك والتى تساعد بالفعل فى تتبع الأسلاك المعطوبة أو فى تدوين أى ملاحظات بالرجوع إلى لون السلك بحيث أن ألوان الأسلاك لم تصنع بطريقة عشوائية بل صنعت لكى تساعد على التمييز بين الأسلاك وبعضها ولكى تستخدم فى استخدامات مختلفة, فمثلاً طبقاً للنظام الأوروبى CEI تم تصنيف الألوان كما بالجدول التالى.

م	الألوان	الاستخدام	التوضيح
١	الأسود والبنى	يستخدم لطرف الكهرباء "الطرف الحى".	شكل رقم (١)
٢	الأحمر, البرتقالى, الرمادى, الأبيض	يستخدم لإغراض عامة "للتوصيل".	شكل رقم (٢)
٣	الأزرق	يستخدم للطرف المتعادل "الطرف الميت".	شكل رقم (٣)
٤	أصفر وأخضر معاً	يستخدم للطرف الأرضى "للحماية".	شكل رقم (٤)



روزتة التيار الكهربى

تستخدم الروزتات الخاصة بلوحات الكنترول لتسهيل عملية التوصيل بين نقاط مختلفة ومتباعدة عن بعضها.

حيث يتم تثبيت الروزتة على لوح التثبيت DIN.



ثم يتم تثبيت السلك بالروزتة باستخدام المفك.



وفى النهاية يتم تركيب غطاء من البلاستك يعمل كعازل على أحدى الجوانب المكشوفة.



فى حالة ربط السلك بالروزتات قد يواجه الشخص مشكلة كبيرة وهى صعوبة أو عدم ضمان تثبيت السلك بطريقة جيدة ومن الطبيعى أن يحدث هذا لعدة أسباب منها:

- صغر حجم لوحة الكنترول.
- السلك من النوع "الشعر".
- صغر مقطع السلك المستخدم "١مم".

فلهذه الأسباب السابق ذكرها يتم استخدام أطراف التثبيت التى تسمى بالترامل "terminals" والموضحة بالشكل التالى



ملاحظة:

- تتوافر الترامل فى الأسواق بأشكال مختلفة حسب الاستخدام فمثلاً عند توصيل أسلاك على الثلاث فازات فأنة يتم استخدام ترامل تسمى "بالشوكة".
- يتم تثبيت طرف السلك بالترامل باستخدام بنسة خاصة, كما بالشكل التالى:



- يتم تحزيم كل مجموعة من الأسلاك معاً حتى يظهر شكل اللوحة فى النهاية بطريقة جيدة بواسطة استخدام أحزمة الأسلاك كما بالشكل التالى:



أعطال جهاز المصدر الكهربى "Power Supply".

لمعرفة أعطال جهاز المصرد الكهربى يجب أولاً فهم فكرة عملة جيداً

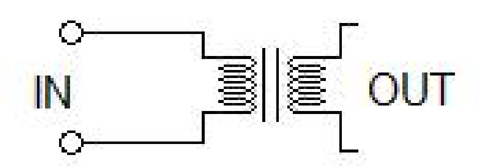


يتكون جهاز المصدر الكهربى من جزئين أساسيين:

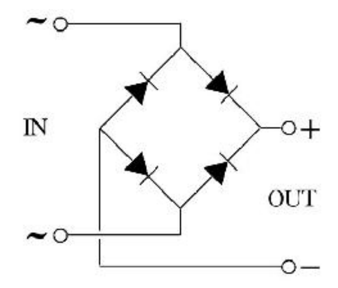
الأول: محول خافض للتيار

الثانى: دائرة لتوحيد التيار

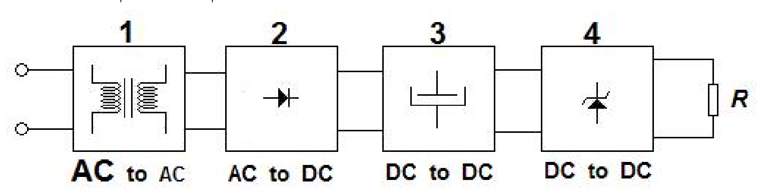
المحول الخافض للتيار يستخدم للتحويل من تيار متردد إلى تيار متردد أخر ولكن أقل قيمة وعادة ما يتم تحويل التيار المتردد من قيمة ٢٢٠ فولت إلى قيمة ٢٤ فولت, كما هو موضح بالشكل التالى.



دائرة توحيد التيار تستخدم للتحويل من تيار متردد إلى تيار مستمر مع الاحتفاظ بنفس قسمة الجهد الأولية فمثلاً يتم تحويل جهد تيار متردد بقيمة ٢٤ فولت إلى جهد تيار مستمر بقيمة ٢٤ فولت, كما هو موضح بالشكل التالى.



يتكون الجهاز من أجزاء أخرى بخلاف الجزئين المذكورين بالأعلى وهم الجزء رقم 3 و 4



الجزء الأول.

محول كهربي خافض للجهد وعادة ما يقوم بتحويل التيار المتردد من ٢٢٠ فولت متردد إلى ٢٤ فولت متردد.

الجزء الثانى.

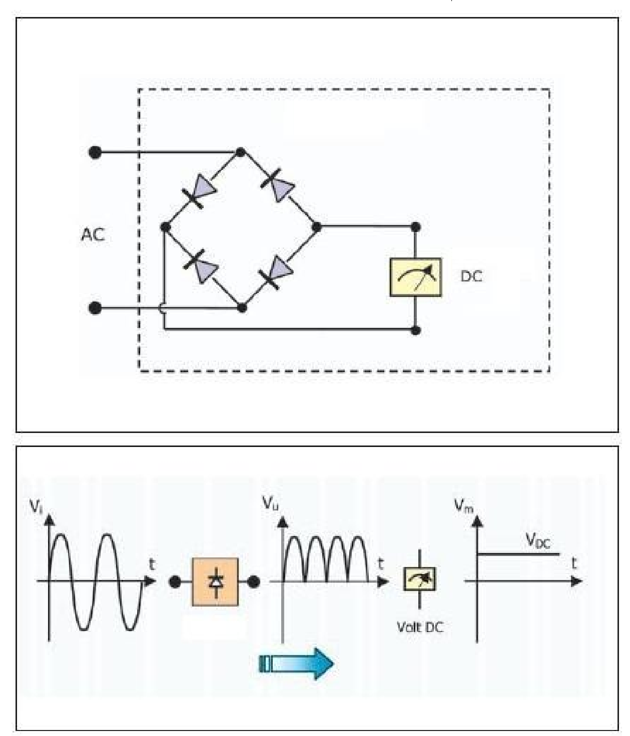
دائرة توحيد كهربية وتستخدم لتحويل قيمة الجهد المتردد إلى نفس القيمة لكن بجهد شبة مستمر.

الجزء الثالث.

مكثف كهربي ويستخدم لمليء الفرغات بالموجة الكهربية لكي يصبح التيار مستمراً.

الجزء الرابع.

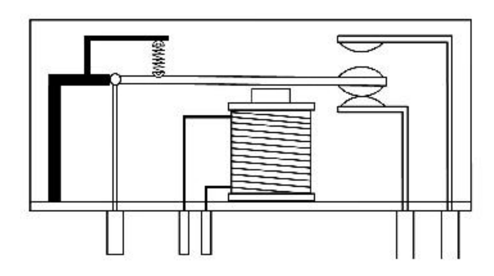
ثنائي زنر "Zener diode"ويستخدم للحصول على مصدر ثابت للتيار المستمر.



المشاكل المتعلقة بجهاز المصدر الكهربى

م	المشكلة	السبب
١	لا يعمل	غير موصل بالكهرباء
		مشكلة بالقاطع "fuse"
٢	اختلاف فى قيمة الجهد	مشكلة بالمحول الخافض
٣	اختلاف فى الموجة المستمرة للجهد	مشكلة بدائرة التوحيد

أعطال الريليه الميكانيكى "Relay".



م	المشكلة	السبب
١	الريليه لا يعمل	مشكلة بملف الريليه
٢	الريليه والحمل لا يعملان	مشكلة بمقعدة التثبيت
٣	الحمل لا يعمل	مشكلة بنقاط الريليه
٤	وضعية نقاط الريليه لا تتغير	مشكلة بالريشة الخارجية

تنفيذ دائرة الألرم ALARM

من الأشياء الضرورية جداً فى الحياة الصناعية التنبية فى حاله وجود أى خلل فى نظام التشغيل فى أى مكينة أو فى خط الإنتاج بالكامل لذالك يعتبر من الأساسى فى تنفيذ أى برنامج أن يتم تصميم برنامج للتنبيه عن الأعطال "alarm" حيث يتكون برنامج التنبية عن الأعطال من جزئين:

الأول – جزء مرئى:

حيث أن فى حاله حدوث العطل أو المشكلة يتم تشغيل لمباتين بشكل متقطع وبطريقة متبادلة لجذب الانتباه إلى حدوث عطل وفى بعض الحالات قد يكتفى البعض باستخدام لمبة واحدة فقط بحيث تضئ بطريقة متقطعة.

ملاحظة:

- فى حالة استخدام لمبتين تتم دائماً البرمجة بحيث يكون زمن إضاءة اللمبة الأولى يساوى زمن أطفاء اللمبة الأخرى وهكذا.
- فى حالة استخدام لمبة واحدة تتم دائماً البرمجة بحيث يكون زمن إضاءة اللمبة أكبر زمن الإطفاء لجذب الانتباه.
- لا يسمح لإى شخص أى أن كان أن يقوم بإطفاء اللمبات يدوياً بحيث أن البرنامج المصمم يقوم بإطفاء اللمبات تلقائياً عند حل المشكلة.
- يتوفر نوع أخر من اللمبات وهى التى تسمى باللمبات المتحركة بحيث يدور جسم عاكس حول اللمبة بواسطة محرك ذات قدرة صغيرة جداً فيبدو وكأن اللمبة هى التى تدور كما هو موضح بالشكل التالى.




الثانى – جزء سمعى:

حيث أن فى حاله حدوث العطل أو المشكلة يتم تشغيل سرينة بشكل متقطع لجذب الانتباه إلى العطل.

ملاحظة:

- فى حالة استخدام السرينة تتم دائماً البرمجة بحيث يكون زمن أطفاء السرينة ضعف زمن التشغيل تقريباً لضمان عدم التأثير السلبى على العمر الأفتراضى للسرينة.
- فقط بالنسبة للسرينة يسمح لإى شخص أى أن كان أن يقوم بإطفاء السرينة يدوياً بحيث أن هذا يشير إلى أنه تم ملاحظة المشكلة وجارى الآن الفحص والتصليح.

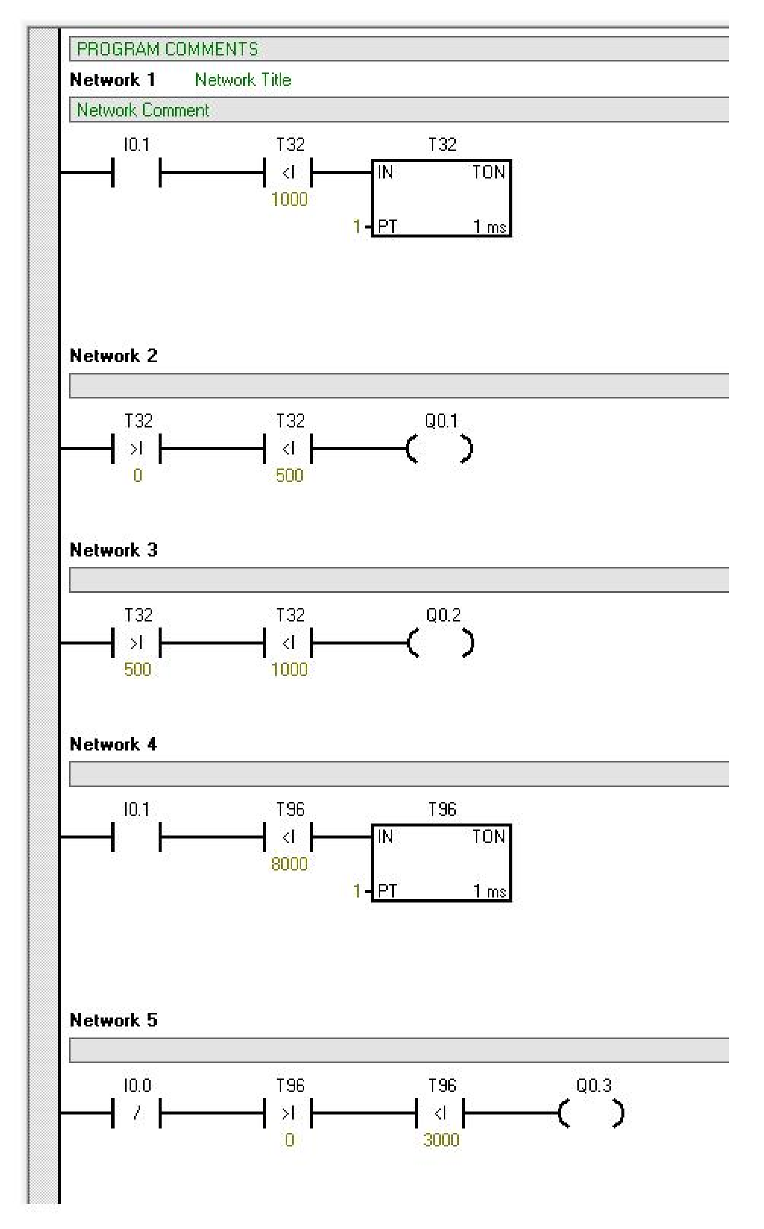


- فى بعض الحالات يتم استبدال السرينة بالجرس الرنان كما الذى بالشكل التالى.



تمرين للتوضيح

برنامج التالى هو لدائرة برمجة منطقية تشير إلى طريقة تصميم برنامج للتنبيه عن الأعطال.



ملاحظة:

قد تتغير بعض الأشياء فى البرنامج حسب المكينة أو حسب نوع العطل.

الجدول:

م	الأسم	العنوان	الشكل
١	لمبة	Q0.1	
٢	لمبة	Q0.2	
٣	سرينة	Q0.3	

الشرح.

الفرع الأول:

عند الضغط على المفتاح I0.1 يبدأ المؤقت الزمنى T32 فى العمل لمدة ثانية واحدة وهكذا.

الفرع الثانى:

تضاء اللمبة الأولى Q0.1 لمدة نصف ثانية فقط.

الفرع الثالث:

تضاء اللمبة الثانية Q0.2 لمدة نصف ثانية أخرى.

الفرع الرابع:

عند الضغط على المفتاح I0.1 يبدأ المؤقت الزمنى T96 فى العمل لمدة ثمان ثوانى وهكذا.

الفرع الخامس:

تعمل السرينة Q0.3 لمدة ثلاث ثوانى وتقف لأرع ثوانى أو إلى أن يتم الضغط على المفتاح I0.0.

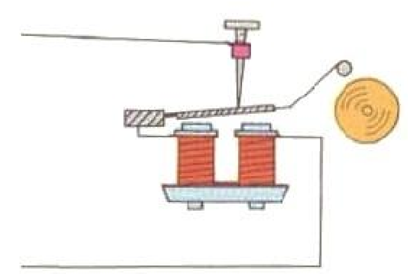
الجرس والسرينة

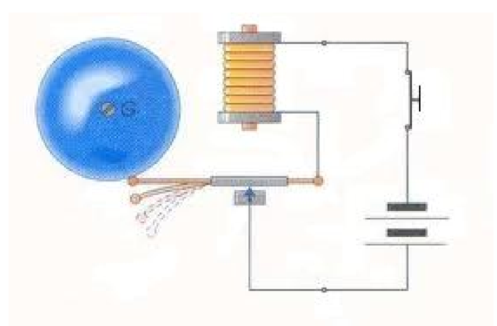
يعتمد كثيراً على حاسة السمع فى لفت الانتباه فلذالك من الضرورى جداً استخدام أدوات صوتية للتنبيه كاستخدام الجرس والسرينة بحيث يعتبر الاثنين من الأشياء الأساسية فى تكوين دائرة الـ alarm.

مكونات الجرس:

تتكون دائرة تصميم الجرس من:

- ملف.
- قلب حديدى.
- ريشة معدنية.
- جسم رنان.
- سوستة.



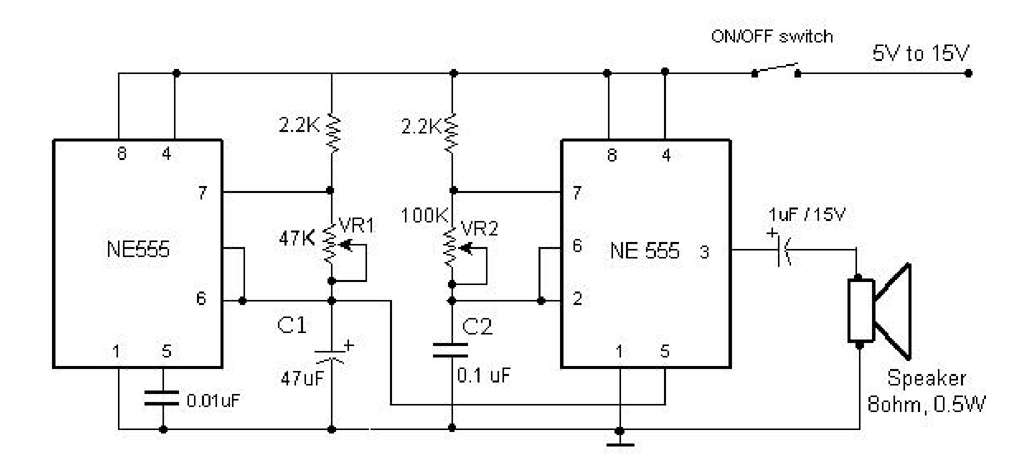


كيف يعمل الجرس:

تعتمد فكرة عمل الجرس على مبدأ أساسى وهو توليد مجال مغنطيسى بواسطة مرور التيار الكهربى بالملف فيقوم المجال المغنطيسى المتولد بتحريك القلب الحديدى ومن ثم يتم سحب الريشة المعدنية نحو الجسم الرنان فتقوم الريشة بقطع مسار التيار بسبب تلك الحركة فينعدم المجال المغنطيسى فتعود الريشة المعدنية إلى طبيعتها بواسطة سحبها فى الاتجاه العكسى بواسطة السوستة المثبتة بالريشة وعندا تعود الريشة المعدنية إلى طبيعتها يتم توصيل التيار الكهربة مرة أخرى وتعود تعمل بنفس الطريقة.

مكونات السرينة:

- دائرتين متكاملتين من نوع NE555.
- مكثف.
- مقاومة متغيرة.



كيف تعمل السرينة:

يستعمل في هذه الدارة دائرتين متكاملتين من نوع NE555 كمتعددين اهتزاز، حيث يمكن التحكم بالتردد بواسطة الدبوس رقم 5.

المكثف C1 يشحن و يفرغ بشكل دوري، و الفولطية عبره تزيد بشكل تدريجي و تنقص بشكل دوري. تنظم هذه الفولطية المختلفة تردد وتتكرر هذه العملية لتسمع صوت رائع مشابه لصفارة الإنذار.

مقاومتي الضبط VR1 و VR2 لتغيير التكرار الدوري للسرينة و لتغيير نغمتها.

المقاومة المتغيرة VR1 تجعلنا نستطيع التحكم بسرعة تغير السرينة من تردد عالي إلى تردد منخفض، اما بالنسبة للمقاومة المتغيرة VR2 فهي تضبط تردد سرينة الإنذار.

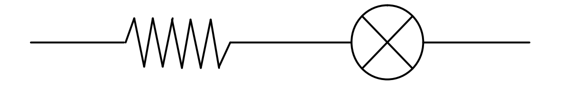
تضبط VR1 و VR2 لتناسب رغبتك.

تعمل الدائرة على فولطية من 5 الى 15 فولت، و مقاومة الجهاز 8 أوم .

مفك التستر "Tester".

يستخدم مفك التستر كاختبار أكيد لوجود الجهد على طرف أى سلك كهربى بحيث تضاء اللمبة المثبتة بالمفتاح فقط فى حاله وجود كهرباء على طرف المفتاح وبشرط أن يكون المفتاح ملامساً للأرضى ويحدث هذا بواسطة تلامس المفتاح مع جسم "يد" الإنسان من الأعلى.





ملاحظة:

- لا يشعر الإنسان بالصدمة الكهربية عند التلامس مع المفتاح وهذا لأن المفتاح يحتوى على مقاومة كبيرة جداً متصلة على التوالى مع اللمبة الداخلية للمفك كما بالشكل التالى من ما يتسبب فى فقدان كبير للجهد ومن ثم يعتبر التيار المار بالدائرة شبه منعدماً.

- قد تصل المقاومة الداخلية المتصلة على التوالى إلى ٣ مليون أوم بينما تصل مقاومة جسم الإنسان إلى ٥٠٠٠ أوم.

لوحة التثبيت "omega bar أو DIN rail".

تستخدم لوحة التثبيت كشيء إساسى لإتمام تثبيت الجهاز بطريقة سلسة ومضمونة لسهولة التوصيل فيما بعد.

أنواع الأجهزة:

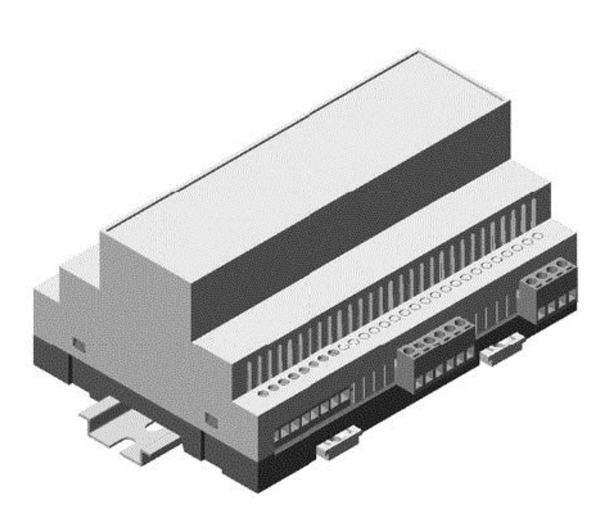
من الأجهزة التى يمكن تثبيتها على لوحات التثبيت (المفتاح الأوتوماتيكى – الأوفرلود – الريليه – التيمر – العداد – الكونتكتور – وحدة البرمجة – وهكذا....)

تتميذ لوحات التثبيت بعدة مميزات منها:

١- تتوفر بمقاسات متعددة لكى يتم استخدامها مع جميع الأجهزة.

٢- تتوفر لوحات التثبيت بمقاسات محددة ومعروفة.

٣- تحتوى اللوحات على فتحات عريضة للتثبيت.

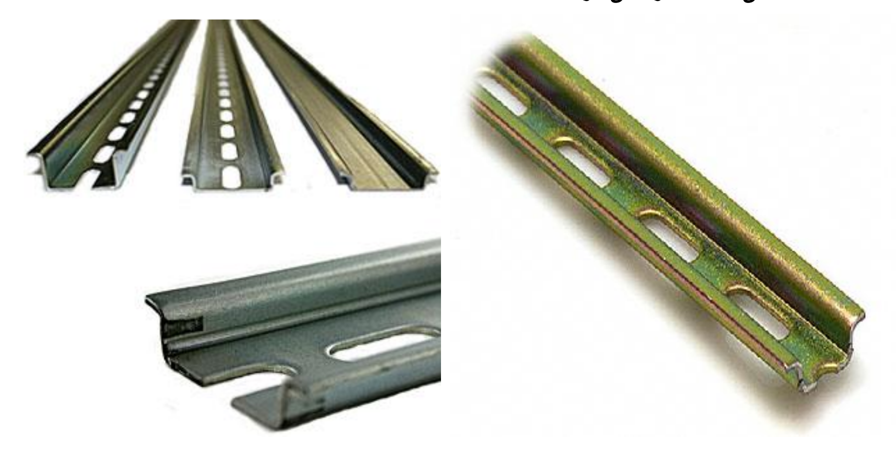


الشكل من الخلف.



ملاحظة:

يجب أن تكون لوحات التثبيت شديدة الاستقامة وأن لا تحتوى على أى أنعواج حتى لا تتسبب فى أى مشاكل أثناء التوصيل أو التثبيت.



مجارى التوصيل "Conduct".

تستخدم مجارى التوصيل لمرور الأسلاك المستخدمة بداخلها وذالك لضمان جودة عالية فى التوصيل ومرونة فى تتبع الأسلاك فيما بعد.



ترقيم الأسلاك **Wire markers**

يعتبر تميز الأسلاك عن بعضا من الأمر الهام جداً خصتاً عندما تكون الدائرة كبيرة وتتكون من أسلاك كثيرة ويتم تميز الأسلاك والمعدات باستخدام الأرقام و الأحرف لكى يصبح من السهل تصميم file excel يحتوى على عمل كل سلك وكل معدة حسب التسلسل المثبت عليها ويفيد هذا جداً فى حالة حدوث أعطال حيث يكون من السهل جداً معرفة بل وتتبع كل شيء بالدائرة دون عناء.



أنواع التلامس مع الكهرباء "Direct & Indirect contact".

ينقسم التلامس بين الإنسان و التيار الكهربى إلى نوعين "تلامس مباشر و تلامس غير مباشر"

النوع الأول:

التلامس المباشر حيث يحدث النلامس بطريقة مباشرة بين جسم الإنسان "اليد مثلاً" ومصدر مباشر للكهرباء مثل (المصدر الرئيسى للكهرباء – السلك الكهربى – مفتاح التحكم) كما هو موضح بالشكل التالى.



النوع الثانى:

التلامس غير المباشر حيث يحدث التلامس بطريقة غير مباشرة بين جسم الإنسان "اليد مثلاً" وجسم المعدة أو المكينة فمثلاً (الجسم الخارجى للمحرك الكهربى – الإطار الخارجى لإى معدة) كما هو موضح بالشكل التالى.



ملاحظة:

- تتوفر فى الأسواق بعض وحدات البرمجة من نوع NPN والتى تتميز بتوصيل القطب السالب على المدخلات بدلاً من توصيل الطرف الموجب وذالك لحماية العامل من التلامس المباشر للتيار.

مفتاح الطوارىء EMERGINCY

يستخدم مفتاح الطوارىء بشكل أساسى فى حاله تصميم أى برنامج حتى وأن كان يبدو بأن لا داعى لاستخدامه لأن فى بعض الحالات الضرورية القصوى قد يحتاج العامل إلى إيقاف كل البرنامج عوضاً عن الاستمرار فى العمل ويحدث هذا فقط فى حالة التأكد من أن استمرار وحدة البرمجة فى العمل قد يتسبب فى ضرر كبير يمكن تجنبه فى حالة توقف البرنامج.

ملاحظة:

يجب أن يكون مفتاح الطوارئ يأتى من خارج وحدة البرمجة وأن لا يكون لوحدة البرمجة تدخل فى طريقة عملة أو فى تغيير الحالة الخاصة بهه.

رسم توضيحى:

الطريقة الصحيحة

PROGRAM COMMENTS
Network 1 Network Title
Network Comment
0.1 I0.0 I1.1 Q0.1
Q0.1

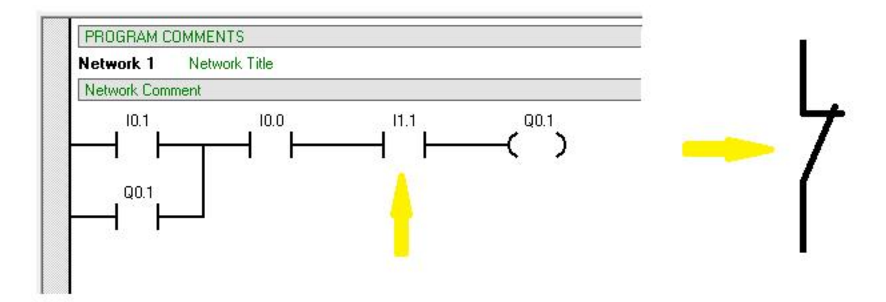
الطريقة الخطاء

PROGRAM COMMENTS
Network 1
I1.1 M0.3
Network 2 Network Title
Network Comment
I0.1 I0.0 M0.3 Q0.1
Q0.1

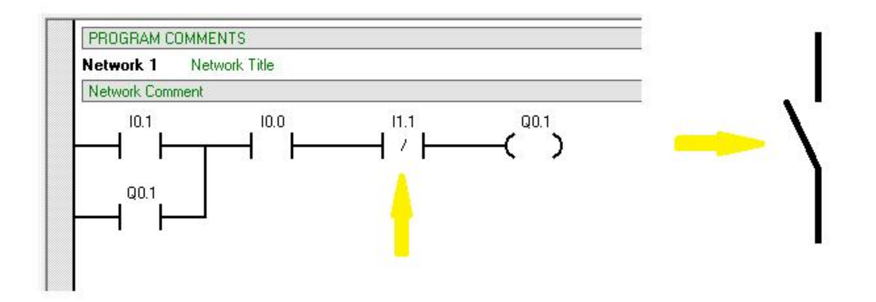
✍ يجب أن يستخدم مفتاح الطوارئ بحيث أن يكون مغلق بالخارج وليس العكس وذالك لأن فى حاله حدوث أى مشكلة "تلف بالمفتاح – عطب بالسلك – انقطاع مصدر الكهرباء" سوف تفهم وحدة البرمجة بأن مفتاح الطوارئ قد تغيرت حالته ففى الحال تقوم وحدة البرمجة بتوقف البرنامج وهذه هى الطريقة الصحيحة بينما إذا تم استخدام مفتاح الطوارئ بحيث أن يكون مفتوح بالخارج ففى حاله انقطاع السلك أو تلف المفتاح لن سيتمكن أحد قط من التحكم بالدائرة.

رسم توضيحى:

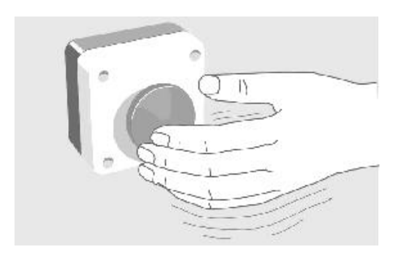
الطريقة الصحيحة



الطريقة الخطاء



شكل عام:





صور لجزء من برنامج خاص بخط أنتاج وقد تم تنفيذه بالفعل باستخدام وحدة برمجة S7-300.

FC1 : Title:

Comment:

Network 1 : Title:

Partenza pistone svuoto magazzino

```
I1.1                    M10.4      I1.4       I1.3       Q5.0
Q5.0
M13.0       M11.3
            (P)
```

Network 2 : Title:

Comment:

```
I1.1.3                               M19.5
                                     (S)
```

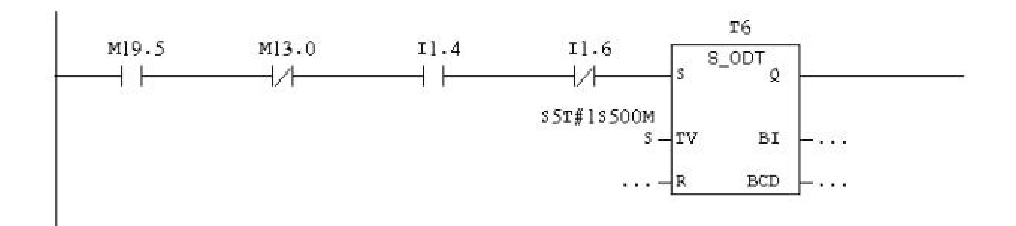
Network 3 : Title:

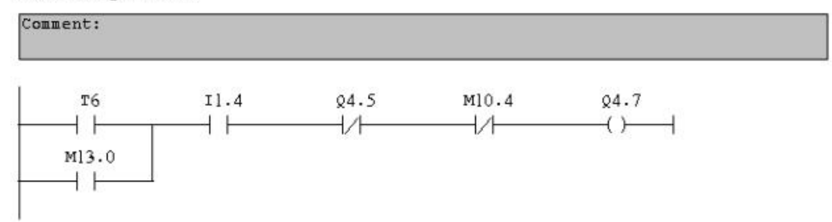
Comment:

```
I1.5        M11.2                    M19.5
            (P)                      (R)
I1.4
```

Network 4 : Title:

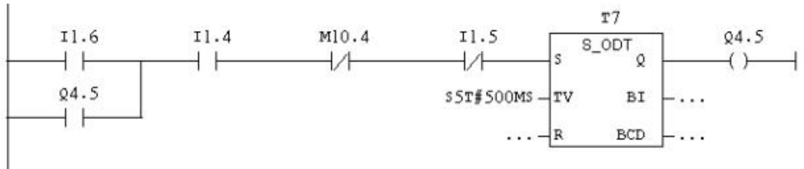
Comment:





Network 6 : Title:





Network 7 : Title:

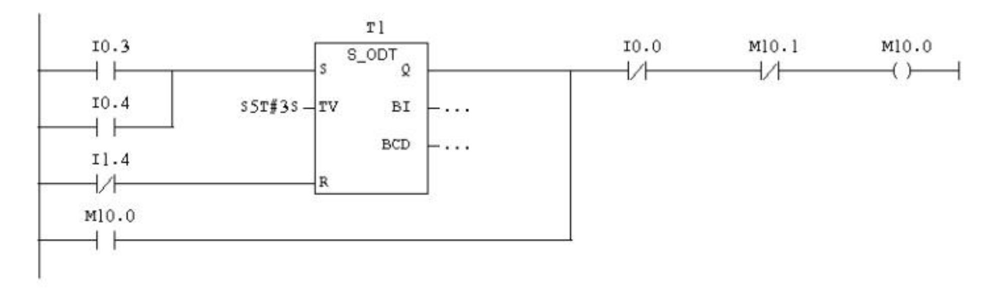


Network 8 : Title:



Network 9 : Title:



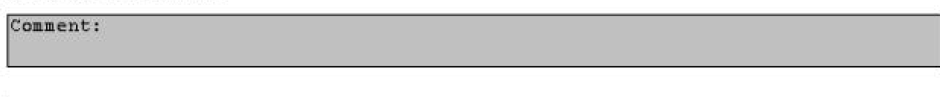


Network 10 : Title:

Comment:

T3
I0.2 S_ODT S Q I0.6 I1.4 M10.1
S5T#1S TV BI ...
I1.4 R BCD ...
M10.1

Network 11 : Title:



M10.0 I1.4 I0.0 M10.4 Q4.2

Network 12 : Title:

Comment:

I0.4 I0.0 I0.6 M10.4 Q4.1
M10.1
T9
M0.3 S_ODT S Q
S5T#1S TV BI ...
... R BCD ...

Network 13 : Title:

Comment:

Q4.3 I0.1 M5.1 M40.4 (S)

Network 14 : Title:

Comment:

M40.4 — M4.4 ( )

Network 15 : Title:

Comment:

T10 / I1.4 / M5.1 — M40.4 (R)

Network 16 : Title:

Comment:

Q4.1 — I0.0 / Q4.3 / I0.0 — I0.5 — T2 S_ODT (S, TV S5T#2S, R, Q, BI ..., BCD ...) / I1.4 (NC) — I1.4 — I0.1 (NC) — Q4.3 ( )

Network 17 : Title:

Comment:

T2 / M10.1 — M5.1 (S)

Network 18 : Title:

Comment:

I1.4 M5.1
(R)
I0.7
I1.3

Network 19 : Title:

Comment:

T4
M5.1 T5 S_ODT
S Q
S5T#100MS TV BI ...
I1.4
R BCD ...

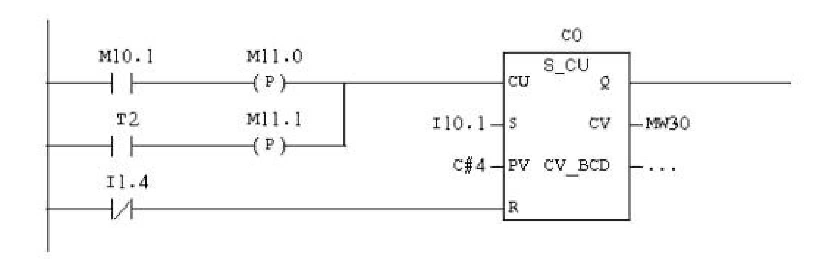
Network 20 : Title:

Comment:

M5.1 T4 Q5.1
( )

Network 21 : Title:

Conta tre allarmi poi ferma tutto I1.4 da cambiare



Network 22 : Title:

Comment:

CMP ==I
MW30 — IN1
4 — IN2
M10.4 ( )

Network 23 : Title:

Comment:

T4 — T5 S_ODT: S, Q
S5T#100MS — TV, BI — ...
... — R, BCD — ...

Network 24 : Title:

Comment:

Q4.1, Q4.0, I1.4, M0.2, M10.1, Q4.0 ( )

Network 25 : Title:

Comment:

Q4.0 — T0 S_ODT: S, Q — M0.2 ( )
S5T#5S — TV, BI — ...
I1.4 — R, BCD — ...

Network 26 : Title:

Per valociazzare il ciclo in caso di presenza di allarmi.

M5.1 M11.4 M13.0 (P) (S) M4.4 M11.5 (P)

Network 27 : Title:

Comment:

I1.4 M13.0 (R) I1.6

Network 28 : Title:

Comment:

T8 I0.2 S_ODT Q5.2 S Q (S) I0.3 S5T#500MS TV BI ... I0.4 I0.1 ... R BCD ... M0.3

Network 29 : Title:

Comment:

I1.4 Q5.2 (R) I0.1 I0.5 M102.0 Q5.2 (P) M0.3 M102.1 (N)

الباب السابع

الطباعة

- أهمية الطبـــــــــــــــــــــــاعة.
- صفحات الطبـــــــــــــــــــــاعة.
- كيفية تحديد صفحة الطبـــــــــــاعة.
- صفحة طبـــــــــــــاعة التعليقات.
- صفحة العرض قبل الطبـــــــــــاعة.
- صفحة خصائص الطبـــــــــــاعة.
- طبـــــــــــــاعة صفحة البيانات.
- المفاتيح المستخدمة فى صفحة الطباعة.
- تمـــــــــــــارين عملية للتوضيح.

أهمية الطباعة:

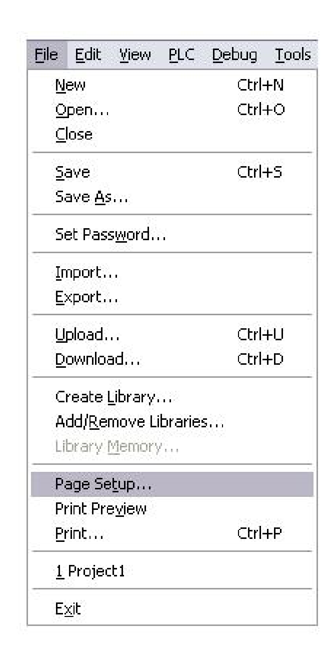
فى بعض الأحيان يكون من الضرورى الاحتفاظ بنسخة من البرنامج لأسباب عدة منها أمكانية قراءة البرنامج لحل الأعطال, وجود نسخة للتحميل مرة أخرى على وحدة البرمجة فى حال تم فقدان البرنامج بسبب انقطاع الكهرباء أو أى مشكلة أخرى حيث يمكن طباعة جميع محتويات البرنامج من رسومات, رموز, تعليقات, متغيرات و صفحات برمجة.

تنقسم الطباعة إلى:

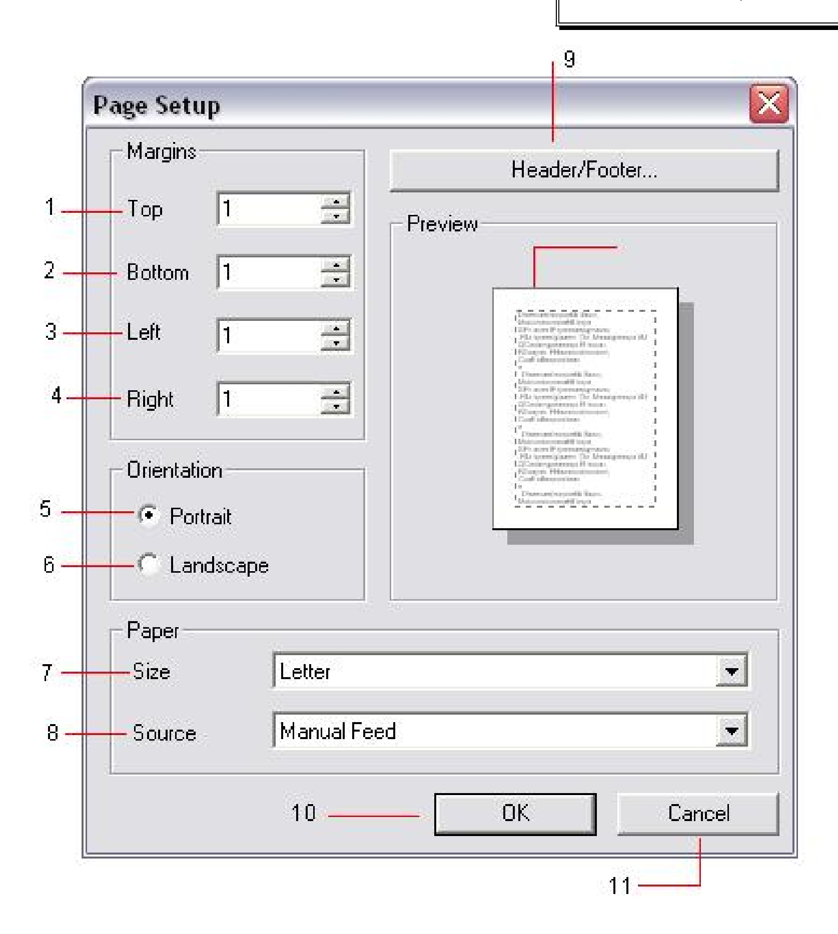
- Page Setup(تحديد الصفحة).
- Print Preview(عرض قبل الطباعة).
- Print(الطبع).

تحديد الصفحة:

يتم الضغط على File ثم Page Setup.



صفحة "تحديد الصفحة":



١- المقصود بكلمة Top هى المسافة الخالية من فوق.

٢- المقصود بكلمة Bottom هى المسافة الخالية من أسفل.

٣- المقصود بكلمة Left هى المسافة الخالية من اليسار.

٤- المقصود بكلمة Right هى المسافة الخالية من اليمين.

٥- المقصود بكلمة Portrait هو الوضع الرأسى للصفحة.

٦- المقصود بكلمة Landscape هو الوضع الأفقى للصفحة.

٧- المقصود بكلمة Size هو حجم الصفحة.

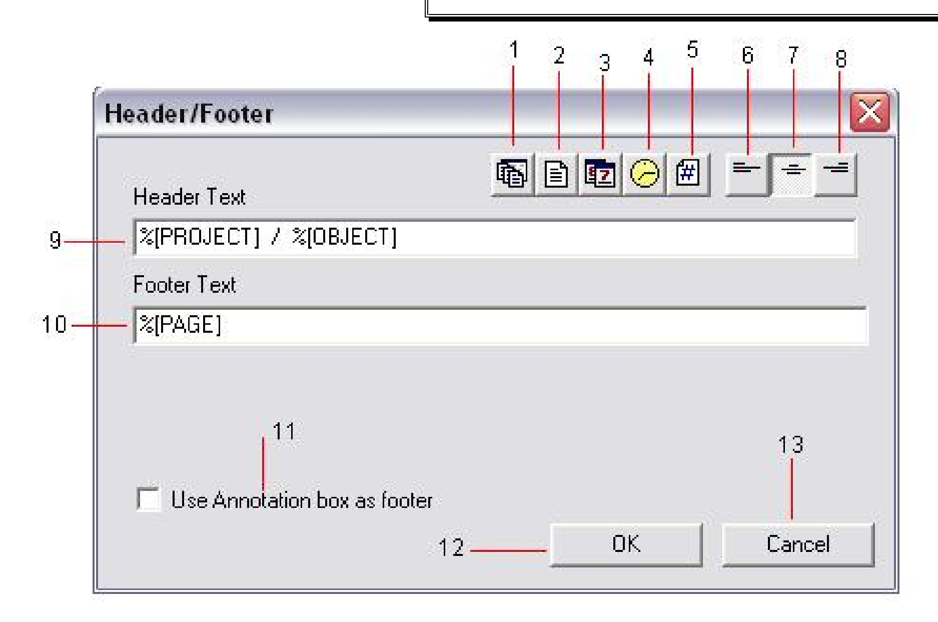
٨- المقصود بكلمة Source هو مصدر الصفحة.

٩- المقصود بكلمة Header/Footer هو الجزء العلوى و السفلى للصفحة.

١٠- المقصود بكلمة Ok هو تفعيل جميع المتغيرات التى تمت وغلق الصفحة أيضاً.

١١- المقصود بكلمة Cancel هو إلغاء التعديلات و غلق الصفحة أيضاً.

صفحة "التعليقات العلوية و السفلية":



١- المقصود بكلمة Project Name هى أسم المشروع.

٢- المقصود بكلمة Object Name هى أسم البرنامج.

٣- المقصود بكلمة Date هى تاريخ طباعة المشروع.

٤- المقصود بكلمة Time هى زمن طباعة المشروع.

٥- المقصود بكلمة Page Number هى رقم الصفحة.

٦- المقصود بكلمة Left Justified هى الكتابة جهة اليسار.

٧- المقصود بكلمة Center Justified هى الكتابة فى المنتصف.

٨- المقصود بكلمة Right Justified هي الكتابة جهة اليمين.

٩- المقصود بكلمة Header Text هو الجزء العلوى للصفحة.

١٠- المقصود بكلمة Footer Text هو الجزء السفلى للصفحة.

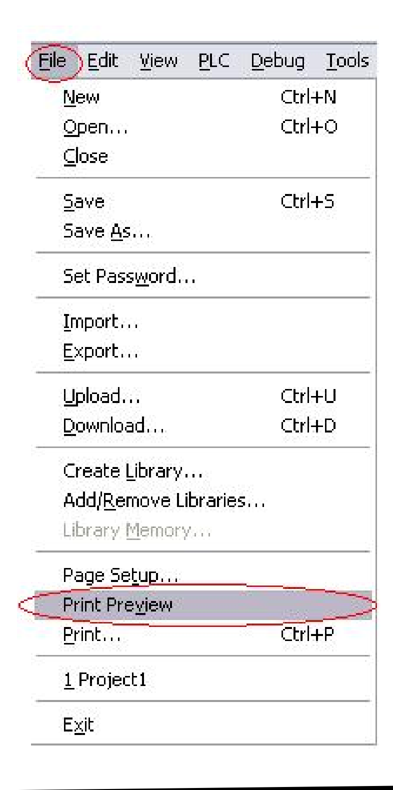
١١- المقصود بكلمة Annotation box هو استخدام جدول الملاحظات كتعليقات فى الجزء السفلى.

١٢- المقصود بكلمة Ok هو تفعيل جميع المتغيرات التى تمت وغلق الصفحة أيضاً.

١٣- المقصود بكلمة Cancel هو إلغاء التعديلات و غلق الصفحة أيضاً.

عرض قبل الطباعة:

يتم الضغط على File ثم Print Preview.



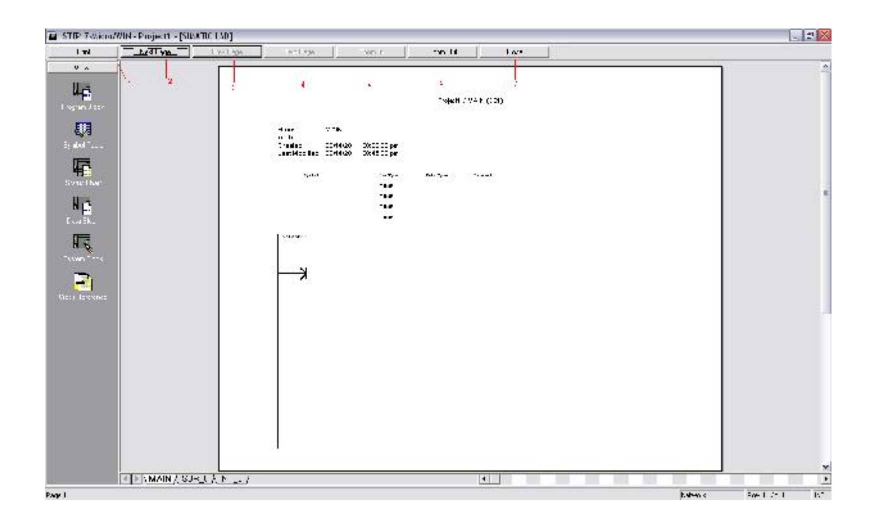
ملاحظة:

- يمكن عرض صفحة البرمجة قبل الطباعة.
- يمكن عرض صفحة الرموز قبل الطباعة.
- يمكن عرض صفحة الحالات قبل الطباعة.
- يمكن عرض صفحة البيانات قبل الطباعة.
- يمكن عرض صفحة نظم البرمجة قبل الطباعة.

✍ يمكن عرض صفحة المرجع قبل الطباعة.

فى حالة عرض البرنامج قبل الطبع قد يظهر البرنامج على عدة أسطر أذا كان البرنامج كبير كما سوف نلاحظ فى أمثلة قادمة.

صفحة الـ "عرض قبل الطبع":



١- المقصود بكلمة Print هى طباعة البرنامج.

٢- المقصود بكلمة Next Page هى عرض الصفحة القادمة.

٣- المقصود بكلمة Prev Page هى عرض الصفحة السابقة.

٤- المقصود بكلمة Two Page هى عرض صفحتان معاً.

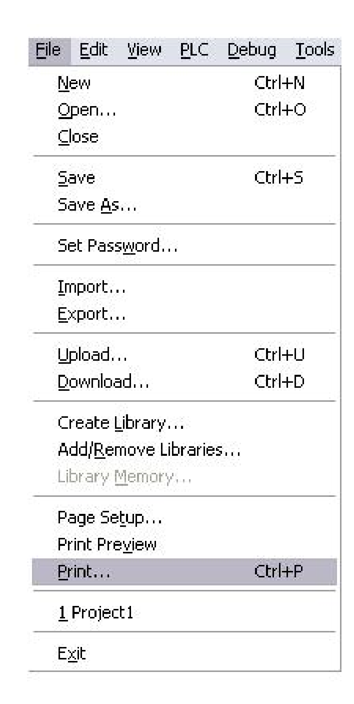
٥- المقصود بكلمة Zoom In هى عرض الصفحة بطريقة أكبر.

٦- المقصود بكلمة Zoom Out هى عرض الصفحة بطريقة أصغر.

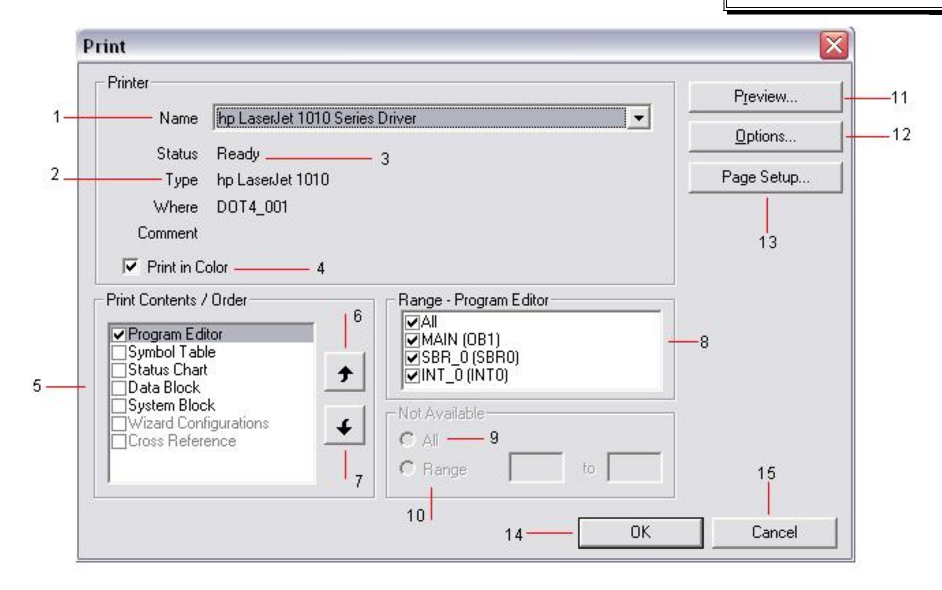
٧- المقصود بكلمة Close هى غلق صفحة الـ Print Preview.

الطبع:

يتم الضغط على File ثم Print.



صفحة " الطبع":



١- المقصود بكلمة Name حيث يتم اختيار الطابعة المراد استخدامه فى حالة وجود أكثر من طابعة.

٢- المقصود بكلمة Status حيث يشير إلى حالة الطابعة.

٣- المقصود بكلمة Type حيث يشير إلى نوع الطابعة التى تم اختيارها

٤- المقصود بكلمة Print in Color هى الطباعة بالألوان خاصة لأن البرنامج يستخدم اللون الأسود و الأخضر و الأحمر.

٥- بواسطة هذه يتم اختيار أى صفح البرمجة يتم طباعتها

٦- بواسطة هذا السهم يتم التحرك إلى أعلى.

٧- بواسطة هذا السهم يتم التحرك إلى أسفل.

٨- المقصود بكلمة Range-Program Editor هو اختيار أى أو كل صفح البرمجة (Main, SBR, INT) .

٩- المقصود بكلمة All هو اختيار جميع أفرع البرنامج للطباعة.

١٠- المقصود بكلمة Range هي اختيار عدد من أفرع البرنامج للطباعة.

١١- المقصود بكلمة Preview هو عرض جميع صفح البرمجة قبل الطباعة.

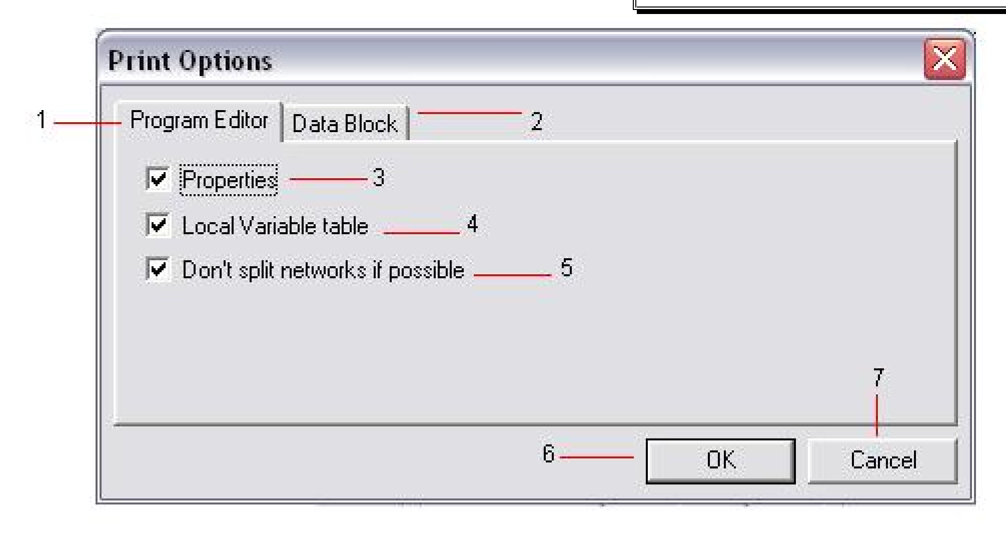
١٢- المقصود بكلمة Option هى صفحة للخيارات كما سوف نرى فيما بعد.

١٣- المقصود بكلمة Page Setup هي تحديد بعض المتغيرات الخاصة بصفحة الطباعة كما سبق وذكرنا.

١٤- المقصود بكلمة Ok هو تفعيل جميع المتغيرات التى نفذت وغلق الصفحة أيضاً.

١٥- المقصود بكلمة Cancel هو إلغاء التعديلات و غلق الصفحة أيضاً.

صفحة "خصائص الطبع":



١- المقصود بكلمة Program Editor هي تعديل خصائص صفح البرمجة.

٢- المقصود بكلمة Data Block هي تعديل خصائص صفحة البيانات.

٣- المقصود بكلمة Properties هو طباعة الخصائص المتعلقة بالبرنامج مثل التاريخ و الساعة و إلخ...

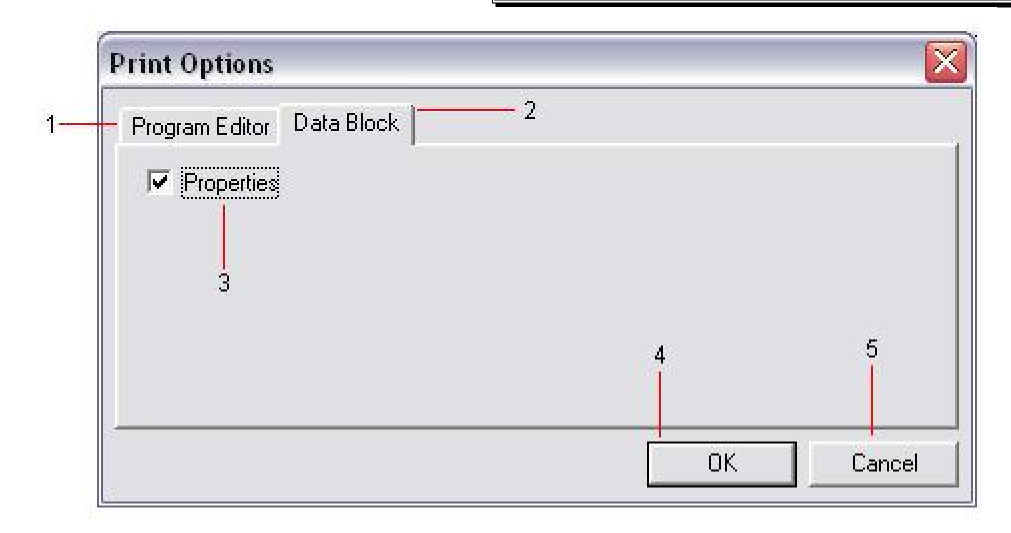
٤- المقصود بكلمة Local Variable Table هي طباعة جدول المتغيرات.

٥- المقصود بها محاولة طباعة فرع البرمجة على سطر واحد قدر المستطاع.

٦- المقصود بكلمة Ok هو تفعيل جميع المتغيرات التى نفذت وغلق الصفحة أيضاً.

٧- المقصود بكلمة Cancel هو إلغاء التعديلات و غلق الصفحة أيضاً.

صفحة "خصائص صفحة البيانات":



١- المقصود بكلمة Program Editor هي تعديل خصائص صفح البرمجة.

٢- المقصود بكلمة Data Block هي تعديل خصائص صفحة البيانات.

٣- المقصود بكلمة Properties هي طباعة الخصائص المتعلقة بصفحة التعليقات و المتغيرات.

٤- المقصود بكلمة Ok هو تفعيل جميع المتغيرات التى نفذت وغلق الصفحة أيضاً.

٥- المقصود بكلمة Cancel هو إلغاء التعديلات و غلق الصفحة أيضاً.

المفاتيح المستخدمة فى الطباعة:



- **:Print**

بالضغط على هذا المفتاح تفتح صفحة "الطبع".

- **:Print Preview**

بالضغط على هذا المفتاح تفتح صفحة "العرض قبل الطباعة".

مثال عملى:

قم بتنفيذ محرك ستار–دلتا:

عدد الدخل	نوع الدخل	أسم الدخل
١	n.c.	I0.0
٢	n.o.	I0.1
٣	n.o.	I1.1
عدد المؤقتات الزمنية	**نوع المؤقتات الزمنية**	**أسم المؤقتات الزمنية**
١	TON	T32
عدد الخرج	**نوع الخرج**	**أسم الخرج**
١	كونتكتور	Q0.0
٢	كونتكتور	Q0.1
٣	كونتكتور	Q0.2

البرنامج

Network 1 Network Title

Network Comment

I0.1 I0.0 I1.1 Q0.0

Q0.0

Network 2

Q0.0 T32

IN TON

2000 PT 1 ms

Network 3

Q0.0 T32 I0.0 Q0.2 I1.1 Q0.1

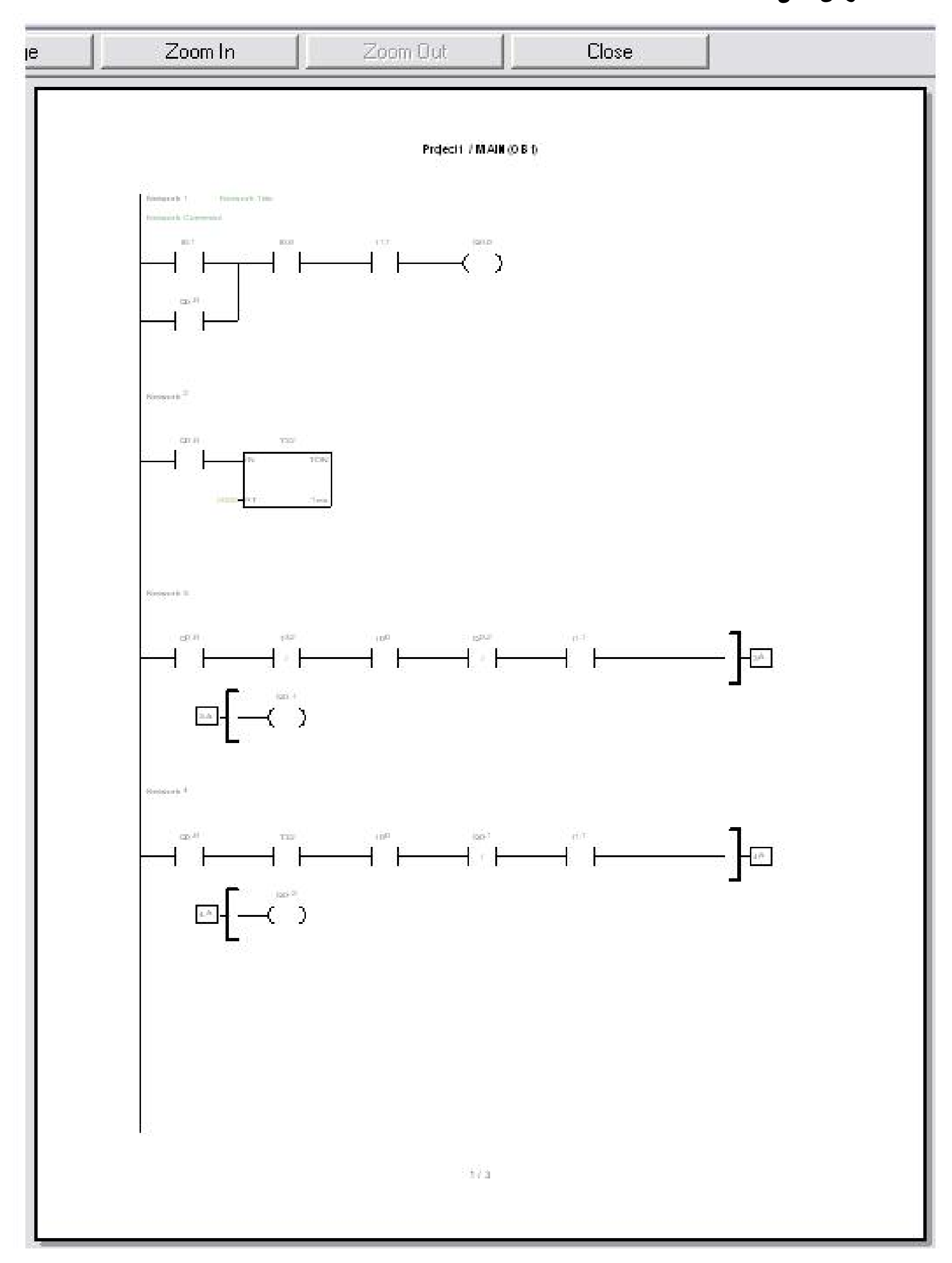
Network 4

Q0.0 T32 I0.0 Q0.1 I1.1 Q0.2

ملاحظة:

الفكرة من البرنامج هى معرفة كيف تتم الطباعة وليس كيفية تنفيذ البرنامج حيث قد سبق توضيح هذا فى الكتابين السابقين.

صفحة "العرض قبل الطباعة":



الباب الثامن

التمارين العملية

- محـــرك يعمل ويقف من مكـــــــــــان واحد
- محـــــــــــرك يعمل ويقف من مكـــــــــانين
- محرك عكس حركة يعمل فى إتجـــــــــــاهين
- لمبتان تعمل بطريقة متبادلة "فلاشـــــــــــــر"
- محـــــــــــــــــرك "ستار – دلتا" عكس حركة
- دائرة إلارم مكونة من لمبتان فلاشر وسرينة تنبيـه
- عــــــــــــــــداد يعمل حتى قيمة كبيرة جداً
- محرك سرعتـــــــــــــــــــــان "داهلاندر"
- عمليات مختلفة بنفس المفتـــــــــــــــــــاح
- محرك عكـــــــــس حركة مع دائــــــرة تنبيه
- التحكم بالمحرك باستخدام التاريـــــخ والساعـــة
- أشـــــــــــــــــارة المـــــــــــــرور
- التحكم بخـــــــــزان بثلاث مستويـــــــــات
- محرك عكس حركة يعمل تلقائياً بعد عودة التيـــار
- التحكم بخـــط تعبـــئة زجاجات بمقاسات مختلفة
- التحكم بثلاث محركات تعمل بطريقة متتـــــالية

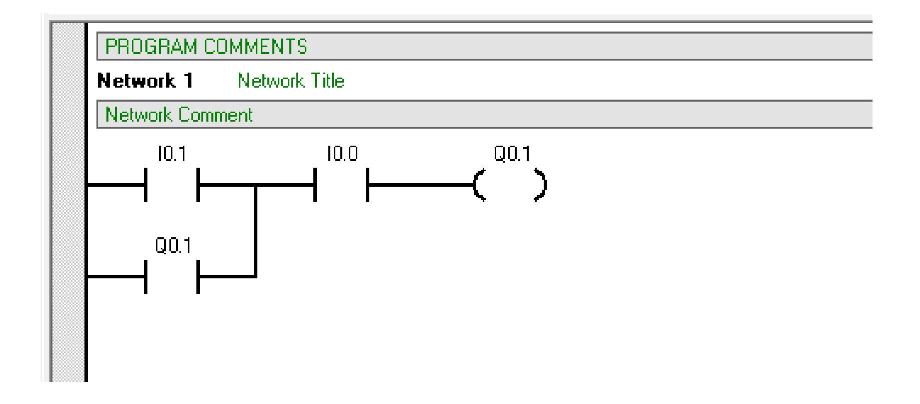
التمرين رقم ١

دائرة التحكم المنطقية لمحرك يعمل ويقف من مكان واحد

الجدول

الاستخدام	الاسم	النوع	م
Start	I0.1	n.o.	١
Stop	I0.0	n.c.	٢
Motor	Q0.1	output	٣

التمرين



الشرح

الفرع الأول.

بالضغط على مفتاح التشغيل I0.1 يعمل المحرك Q0.1 وبالضغط على المفتاح I0.0 يقف المحرك.

التمرين رقم ٢

دائرة التحكم المنطقية لمحرك يعمل من مكانين ويقف من مكانين.

الجدول

م	النوع	الاسم	الاستخدام
	n.c.	I0.0	Stop1
١	n.o.	I0.1	Start1
٢	n.o.	I0.2	Start2
٣	n.c.	I2.3	Stop2
٤	output	Q0.6	Motor

التمرين

PROGRAM COMMENTS
Network 1 Network Title
Network Comment
I0.1 I0.0 I2.3 Q0.6
I0.2
Q0.6

الشرح

الفرع الأول.

بالضغط على مفتاح التشغيل I0.1 أو I0.2 يعمل المحرك Q0.1 وبالضغط على المفتاح I0.0 أو I2.3 يتوقف

التمرين رقم ٣

دائرة التحكم المنطقية لمحرك عكس حركة يعمل فى اتجاهين.

الجدول

م	النوع	الأسم	الاستخدام
١	n.o.	I0.1	Start-R
٢	n.o.	I0.2	Start-L
٣	n.c.	I0.0	Stop
٤	output	Q0.1	Motor-R
٥	output	Q0.2	Motor-L

التمرين

PROGRAM COMMENTS

Network 1 Network Title

Network Comment

I0.1 I0.0 Q0.2 Q0.1

Q0.1

Network 2 Network Title

Network Comment

I0.2 I0.0 Q0.1 Q0.2

Q0.2

الشرح

الفرع الأول.

بالضغط على مفتاح التشغيل I0.1 يعمل المحرك فى الاتجاه اليمين Q0.1 وبالضغط على المفتاح I0.0 يتوقف المحرك عن الدوران.

الفرع الثانى.

بالضغط على مفتاح التشغيل I0.2 يعمل المحرك فى الاتجاه اليسار Q0.2 وبالضغط على المفتاح I0.0 يتوقف المحرك عن الدوران.

التمرين رقم ٤

دائرة التحكم المنطقية لمحرك عكس حركة يعمل فى اتجاهين حتى نهاية المشوار الخاص بكل اتجاه.

الجدول

م	النوع	الاسم	الاستخدام
١	n.o.	I0.1	Start-R
٢	n.o.	I0.2	Start-L
٣	n.c	I1.1	L.S.-R
٤	n.c	I1.2	L.S.-L
٥	n.c.	I0.0	Stop
٦	output	Q0.1	Motor-R
٧	output	Q0.2	Motor-L

التمرين

PROGRAM COMMENTS

Network 1 Network Title

Network Comment

I0.1 I0.0 I1.1 Q0.2 Q0.1

Q0.1

Network 2 Network Title

Network Comment

I0.2 I0.0 I1.2 Q0.1 Q0.2

Q0.2

الشرح

الفرع الأول.

بالضغط على مفتاح التشغيل I0.1 يعمل المحرك فى الاتجاه اليمين Q0.1 ويتوقف المحرك عن الحركة عند الوصول إلى مفتاح نهاية المشوار I1.1 أو بالضغط على مفتاح الإيقاف I0.0

الفرع الثانى.

بالضغط على مفتاح التشغيل I0.2 يعمل المحرك فى الاتجاه اليسار Q0.2 ويتوقف المحرك عن الحركة عند الوصول إلى مفتاح نهاية المشوار I1.2 أو بالضغط على مفتاح الإيقاف I0.0

التمرين رقم ٥

دائرة التحكم المنطقية للمبتان تعمل بطريقة متبادلة "فلاشر".

الجدول

م	النوع	الاسم	الاستخدام
١	n.o.	I0.1	Selector
٢	output	Q0.1	Motor-R
٣	output	Q0.2	Motor-L

التمرين

Due lampade che lampeggiano

Network 1 Network Title

Network Comment

I0.1 T32 <=I 1000 T32 IN TON 1 PT 1 ms

Network 2

I0.1 T32 <=I 500 Q0.1

Network 3

T32 >=I 500 Q0.2

الشرح

الفرع الأول.

بالضغط على المفتاح I0.1 يعمل المؤقت الزمنى T32 إلى أن تتعدى قيمة المؤقت الزمنى الـ ١٠٠٠ مللى ثانية.

الفرع الثانى.

تضيء اللمبة الأولى Q0.1 عندما يكون زمن المؤقت الزمنى أقل من نصف ثانية.

الفرع الثالث.

تضيء اللمبة الثانية Q0.2 عندما يكون زمن المؤقت الزمنى أكثر من نصف ثانية.

التمرين رقم ٦

دائرة التحكم المنطقية لمحرك "ستار – دلتا".

الجدول

م	النوع	الاسم	الاستخدام
١	n.o.	I0.1	Start
٢	n.c.	I0.0	Stop
٣	output	Q0.0	Main
٤	output	Q0.1	Star
٥	output	Q0.2	Delta

التمرين

PROGRAM COMMENTS

Network 1 Network Title

STELLA

I0.1 I0.0 T32 / Q0.2 / Q0.1 ( )

Q0.1

Network 2

MAIN

Q0.1 I0.0 Q0.0 ( )

Q0.0

Network 3

TRIANGOLO

Q0.0 T32

IN TON

4000 PT 1 ms

Network 4

T32 Q0.1 / Q0.2 ( )

الشرح

الفرع الأول.

بالضغط على مفتاح التشغيل I0.1 يعمل الكونتكتور "ستار" Q0.1 وبالضغط على مفتاح الإيقاف I0.0 يتم فصل كونتكتور الستار.

الفرع الثاني.

بعد تشغيل كونتكتور الستار يعمل الكونتكتور الرئيسى Q0.0 وبالضغط على مفتاح الإيقاف I0.0 يتم فصل الكونتكتور الرئيسى.

الفرع الثالث.

يعمل المؤقت الزمنى T32 للتحكم بكونتكتور الـ "ستار" و الـ "دلتا".

الفرع الرابع.

بعد مرور زمن محدد مسبقاً يعمل كونتكتور الـ "دلتا".

التمرين رقم ٧

دائرة التحكم المنطقية لمحرك "ستار – دلتا" عكس حركة.

الجدول

م	النوع	الاسم	الاستخدام
١	n.o.	I0.1	Start
٢	n.c.	I0.0	Stop
٣	output	Q0.6	Main-R
٤	output	Q0.7	Main-L
٥	output	Q0.1	Star
٦	output	Q0.2	Delta

الشرح

الفرع الأول.

بالضغط على مفتاح التشغيل I0.1 يعمل المحرك ستار دلتا فى الاتجاه اليمين Q0.6 وبالضغط على مفتاح الإيقاف I0.0 يتوقف المحرك.

الفرع الثانى.

بالضغط على مفتاح التشغيل I0.2 يعمل المحرك ستار دلتا فى الاتجاه اليسار Q0.7 وبالضغط على مفتاح الإيقاف I0.0 يتوقف المحرك.

وباقى الأفرع كما بالمثال السابق.

التمرين

PROGRAM COMMENTS

Network 1

I0.1 I0.0 Q0.7 Q0.6

Q0.6

Network 2

I0.2 I0.0 Q0.6 Q0.7

Q0.7

Network 3 Network Title

STELLA

Q0.6 T32 Q0.2 Q0.1

Q0.7

Network 4

TRIANGOLO

Q0.6 T32

IN TON

Q0.7 4000 PT 1 ms

Network 5

T32 Q0.1 Q0.2

التمرين رقم ٨

دائرة التحكم المنطقية لدائرة الإرم تتكون من لمبتان فلاشر وسرينة تنبيه.

الجدول

م	النوع	الاسم	الاستخدام
١	n.o.	I0.1	Selector
٢	output	Q0.1	Lamp1
٣	output	Q0.2	Lamp2
٤	output	Q0.3	horn

الشرح

الفرع الأول.

بالضغط على المفتاح I0.1 يعمل المؤقت الزمنى T32 إلى أن تتعدى قيمة المؤقت الزمنى الـ ١٠٠٠ مللى ثانية.

الفرع الثانى.

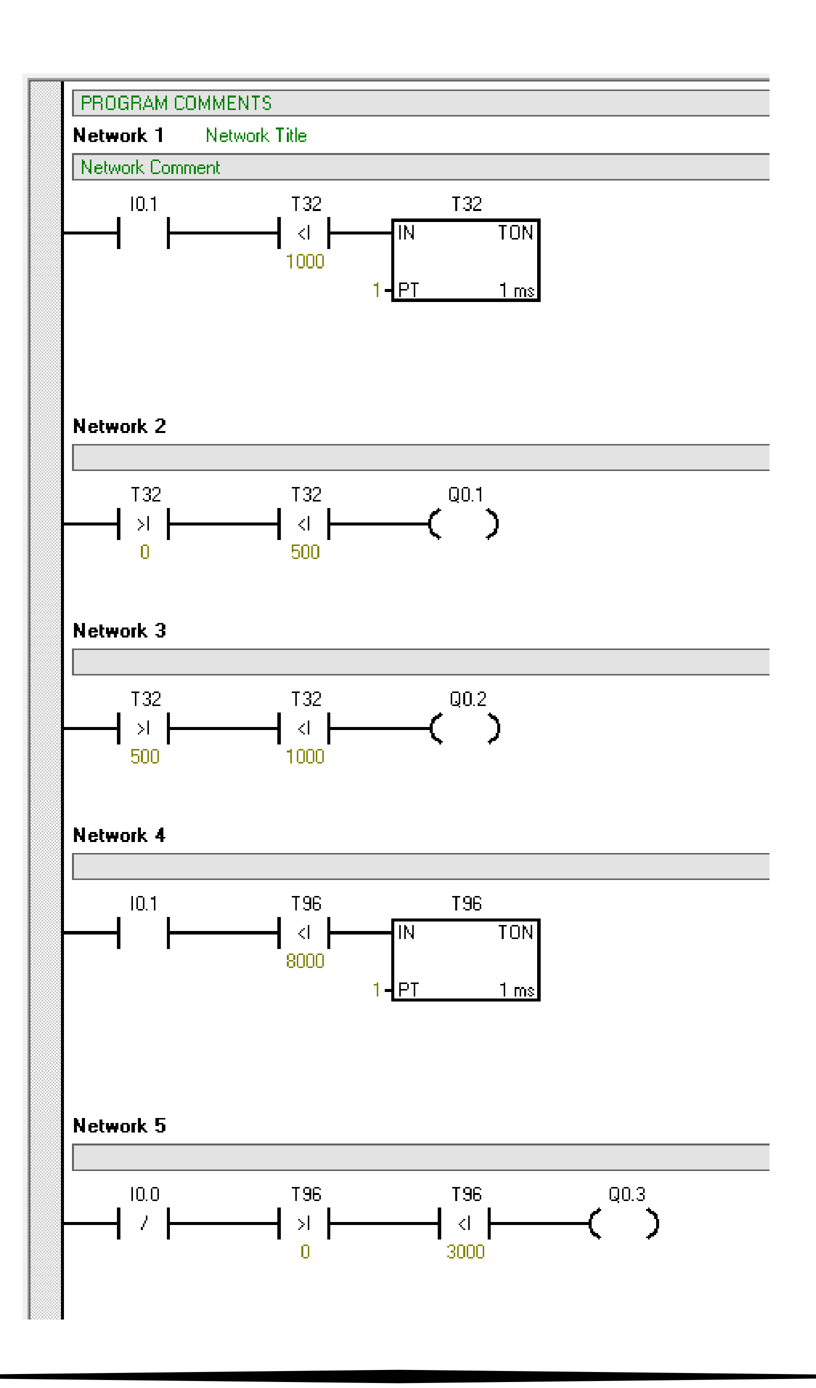
تضيء اللمبة الأولى Q0.1 عندما يكون زمن المؤقت الزمنى أقل من نصف ثانية.

الفرع الثالث.

تضيء اللمبة الثانية Q0.2 عندما يكون زمن المؤقت الزمنى أكثر من نصف ثانية.

وبنفس الطريقة يتم التحكم بالسرينة Q0.3 أيضاً.

التمرين


PROGRAM COMMENTS
Network 1
Network Title
Network Comment
I0.1
T32
<I
1000
T32
IN
TON
1
PT
1 ms
Network 2
T32
>I
0
T32
<I
500
Q0.1
Network 3
T32
>I
500
T32
<I
1000
Q0.2
Network 4
I0.1
T96
<I
8000
T96
IN
TON
1
PT
1 ms
Network 5
I0.0
/
T96
>I
0
T96
<I
3000
Q0.3

التمرين رقم ٩

دائرة التحكم المنطقية لتصميم عداد يعمل حتى قيمة كبيرة جداً.

الجدول

الاستخدام	الاسم	النوع	م
Start	I0.1	n.o.	١
Reset	I0.0	n.o.	٢
Lamp	Q0.1	output	٣

الشرح

الفرع الأول.

بالضغط على المفتاح "الحساس" I0.1 يعمل العداد التصاعدى بدايتاً من الصفر.

الفرع الثانى.

بالضغط على المفتاح I0.0 تعود قيمة الذاكرة الخاصة بالعداد التصاعدى إلى صفر.

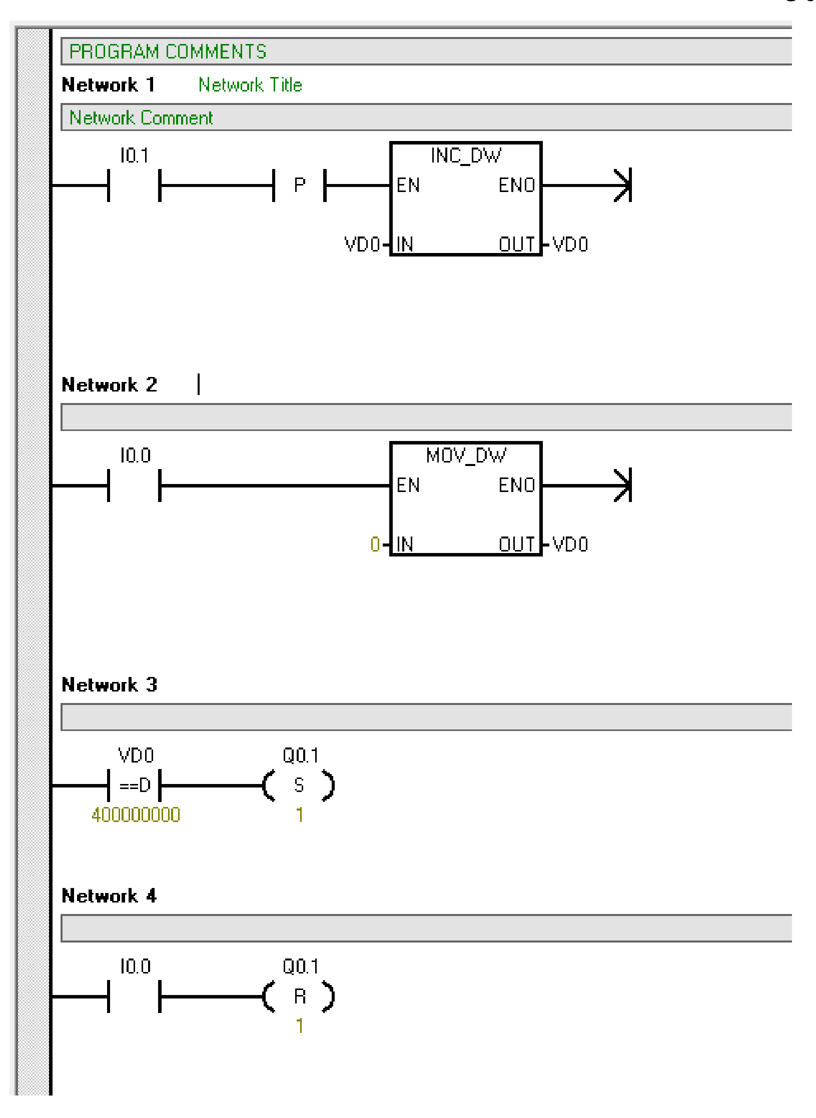
الفرع الثالث.

تضيء اللمبة Q0.1 عندما تعادل قيمة العداد التصاعدى الرقم المحدد.

الفرع الرابع.

تطفئ اللمبة Q0.1 عندما يتم الضغط على مفتاح الإيقاف I0.0.

التمرين


PROGRAM COMMENTS
Network 1
Network Title
Network Comment
I0.1
P
INC_DW
EN
ENO
VD0
IN
OUT
VD0
Network 2
I0.0
MOV_DW
EN
ENO
0
IN
OUT
VD0
Network 3
VD0
==D
400000000
Q0.1
S
1
Network 4
I0.0
Q0.1
R
1

التمرين رقم ١٠

دائرة التحكم المنطقية للتحكم بمحرك سرعتان "داهلاندر".

الجدول

م	النوع	الاسم	الاستخدام
١	n.o.	I0.1	Start
٢	n.c.	I0.0	Stop
٣	output	Q0.1	Low speed
٤	output	Q0.2	High speed
٥	output	Q0.3	Star

الشرح

الفرع الأول.

يبدأ المؤقت الزمنى بالعمل مع السرعة البطيئة.

الفرع الثانى.

بالضغط على المفتاح I0.1 يعمل المحرك بالسرعة البطيئة Q0.1 ويتوقف عند الضغط على مفتاح الإيقاف I0.0.

الفرع الثالث.

بعد مرور الزمن المحدد مسبقاً يعمل المحرك بالسرعة السريعة بواسطة الكونتكتور Q0.3 ستار.

الفرع الرابع.

بعد عمل الكونتكتور Q0.3 يعمل أيضاً الكونتكتور Q0.2 الخاص بالسرعة السريعة.

التمرين

PROGRAM COMMENTS

Network 1

Q0.1 — T32 (IN, TON), 7000 — PT, 1 ms

Network 2 Network Title

Network Comment

I0.1 — I0.0 — Q0.2 / — Q0.3 / — T32 / — (Q0.1)

Q0.1

Network 3

T32 — I0.0 — Q0.1 / — (Q0.3)

Q0.3

Network 4

Q0.3 — Q0.1 / — (Q0.2)

التمرين رقم ١١

دائرة التحكم المنطقية لمحركين حيث بالضغط على المفتاح للمرة الأولى يعمل المحرك الأول وبالضغط على نفس المفتاح للمرة الثانية يعمل المحرك الثانى وبالضغط على المفتاح للمرة الثالثة يقف المحرك الأول وعند الضغط للمرة الرابعة يقف المحرك الثانى.

الجدول

م	النوع	الاسم	الاستخدام
١	n.o.	I0.1	Start
٢	n.o.	I0.0	Stop
٣	output	Q0.1	Motor1
٤	output	Q0.2	Motor2

الشرح

- الفرع الأول.

بالضغط على المفتاح I0.1 يعمل العداد ويعود إلى الصفر تلقائياً أو بالضغط على مفتاح الإيقاف I0.0.

- الفرع الثانى.

يعمل المحرك Q0.1 عندما تصبح قيمة العداد ١ أى عند الضغطة الأولى على المفتاح.

- الفرع الثالث.

يعمل المحرك Q0.2 عندما تصبح قيمة العداد ٢ أى عند الضغطة الثانية على المفتاح.

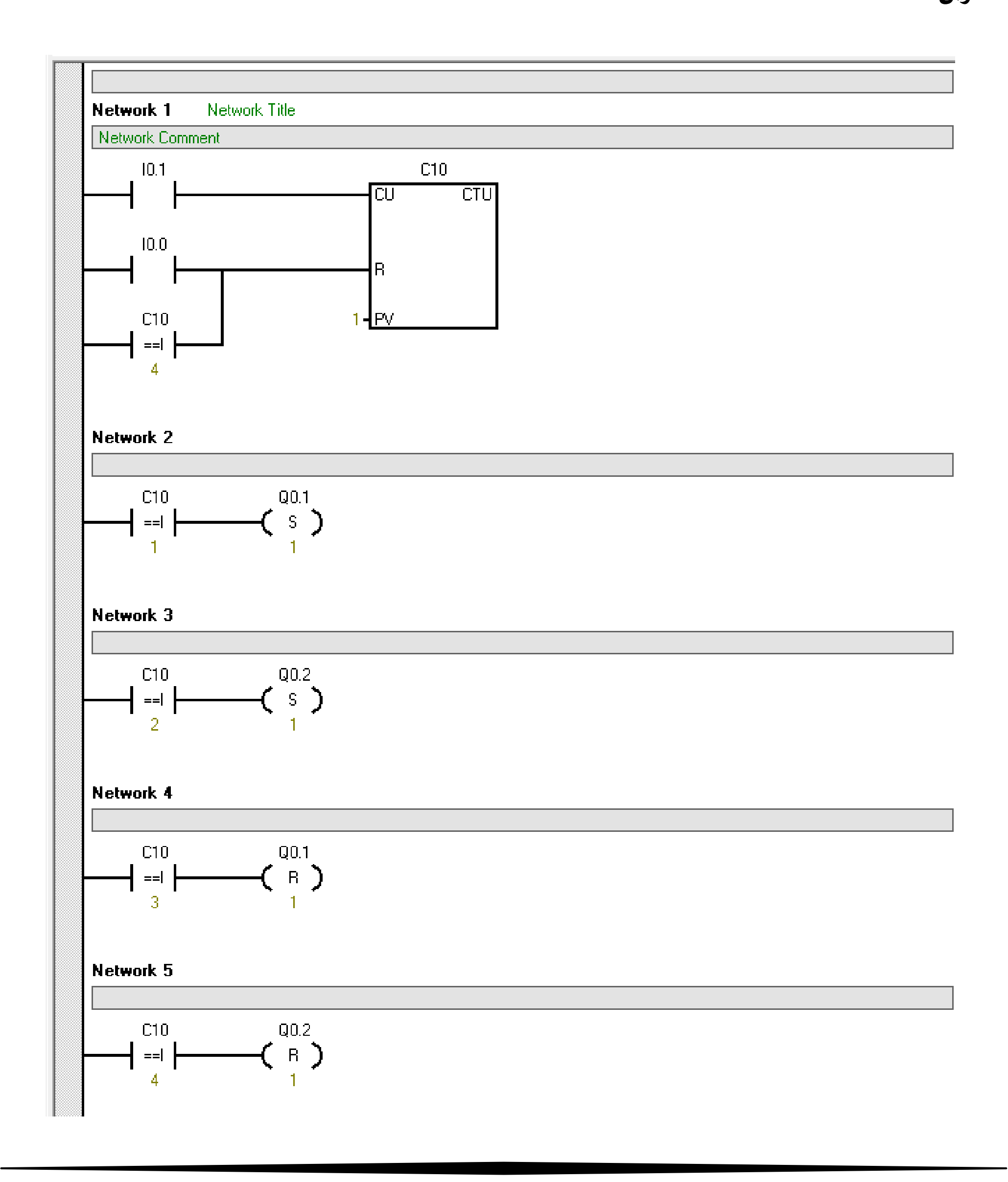
- الفرع الرابع.

يقف المحرك Q0.1 عندما تصبح قيمة العداد ٣ أى عند الضغطة الثالثة على المفتاح.

الفرع الخامس.

يقف المحرك Q0.2 عندما تصبح قيمة العداد ٤ أى عند الضغطة الرابعة على المفتاح.

التمرين



التمرين رقم ١٢

دائرة التحكم المنطقية لمحرك عكس حركة حيث يعمل الأتجاة المحدد بالضغط على المفتاح الخاص بة وعندما يعمل أى اتجاه تعمل تلقائياً دائرة فلاشر مكونة من لمبتين يعملان بالتبادل لمدة ثانية لكل منهما وتتكرر العملية حتى يتوقف المحرك عن العمل ويعمل أيضاً جرس تحذير لمدة ثانية واحدة فقط.

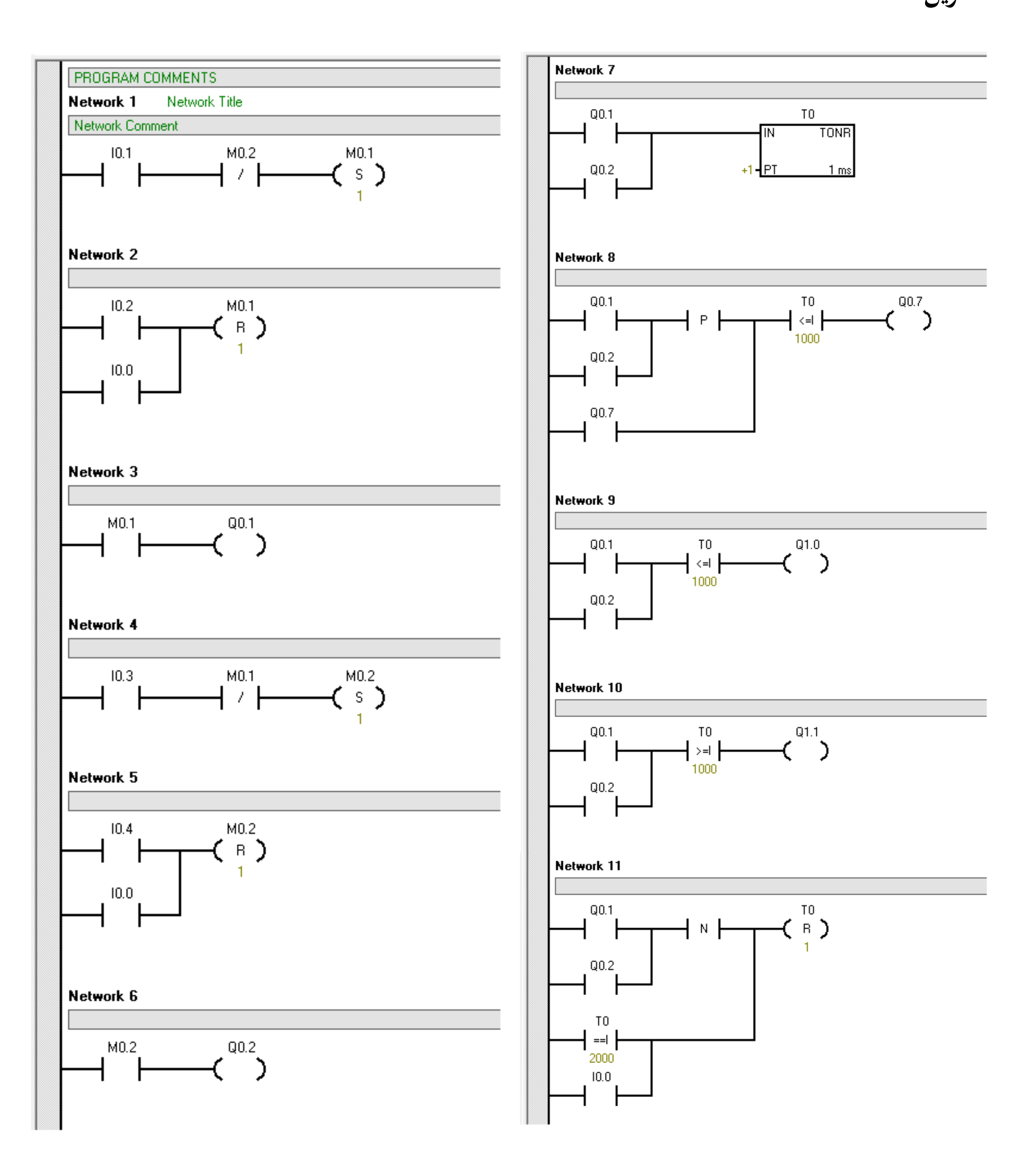
الجدول

م	النوع	الاسم	الاستخدام
١	n.o.	I0.1	Start
٢	n.o.	I0.2	Stop
٣	n.o.	I0.3	Start
٤	n.o.	I0.4	Stop
٥	output	Q0.1	Motor-R
٦	output	Q0.2	Motor-L
٧	output	Q0.7	Horn
٨	output	Q1.0	Lamp1
٩	output	Q1.1	Lamp2

ملاحظة:

لقد تم استخدام الـ Set/Reset وكذالك المؤقت الزمنى TONR لأنهما يحتفظا بحالتهما وبقيمهما ففى حالة انقطاع الكهرباء بعد أنقطعها يعمل التمرين تلقائياً من نفس النقطة التى توقف عندها.

التمرين



PROGRAM COMMENTS
Network 1
Network Title
Network Comment
I0.1
M0.2
M0.1
S
1
Network 2
I0.2
M0.1
R
1
I0.0
Network 3
M0.1
Q0.1
Network 4
I0.3
M0.1
M0.2
S
1
Network 5
I0.4
M0.2
R
1
I0.0
Network 6
M0.2
Q0.2
Network 7
Q0.1
T0
IN
TONR
Q0.2
+1
PT
1 ms
Network 8
Q0.1
P
T0
<=I
1000
Q0.7
Q0.2
Q0.7
Network 9
Q0.1
T0
<=I
1000
Q1.0
Q0.2
Network 10
Q0.1
T0
>=I
1000
Q1.1
Q0.2
Network 11
Q0.1
N
T0
R
1
Q0.2
T0
==I
2000
I0.0

الشرح

- الفرع الأول. بالضغط على المفتاح I0.1 يعمل الريليه M0.1
- الفرع الثانى. يتوقف الريليه M0.1 بسبب مفتاح نهاية المشوار I0.2 أو مفتاح الإيقاف I0.0
- الفرع الثالث. عندما يعمل الريليه M0.1 يعمل أيضاً المحرك Q0.1
- الفرع الرابع. بالضغط على المفتاح I0.2 يعمل الريليه M0.2
- الفرع الخامس. يتوقف الريليه M0.2 بسبب مفتاح نهاية المشوار I0.4 أو مفتاح الإيقاف I0.0
- الفرع السادس. عندما يعمل الريليه M0.2 يعمل أيضاً المحرك Q0.2
- الفرع السابع. فى حالة عمل المحرك فى أى اتجاه يعمل أيضاً المؤقت الزمنى T32
- الفرع الثامن. تعمل السرينة التحذيرية لمدة ثانية واحدة فقط.
- الفرع التاسع. تضئ اللمبة الأولى Q1.0 لمدة نصف ثانية ثم تطفئ لمدة نصف ثانية أخرى وهكذا.
- الفرع العاشر. تضئ اللمبة الثانية Q1.1 لمدة نصف ثانية ثم تطفئ لمدة نصف ثانية أخرى وهكذا.
- الفرع الحادى عشر. يرجع المؤقت الومنى إلى الصفر لكى يعمل من جديد بعد مرور ثانيتين من التشغيل أو عند توقف المحرك.

التمرين رقم ١٣

دائرة التحكم المنطقية لتنفيذ دائرة بحيث يتم التحكم بالمحرك باستخدام التاريخ والساعة.

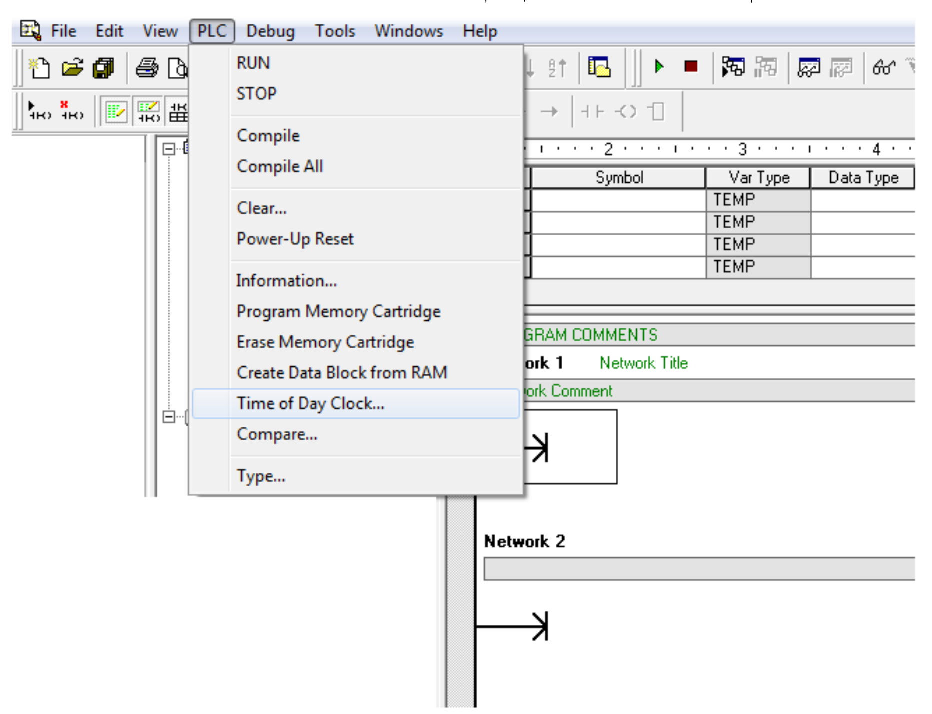
الجدول

م	النوع	الاسم	الاستخدام
١	output	Q0.1	Motor

خطوات ضبط الساعة.

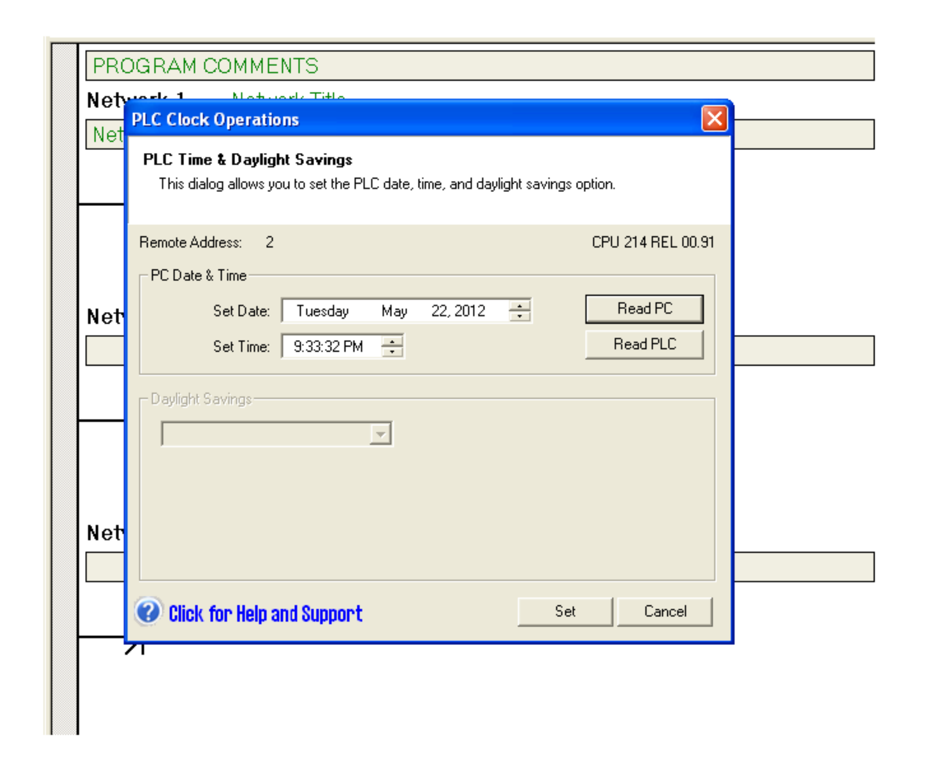
ملاحظة:

لضبط الساعة يتم اختيار قائمة PLC ثم يتم الضغط على Time of Day.



ملاحظة:

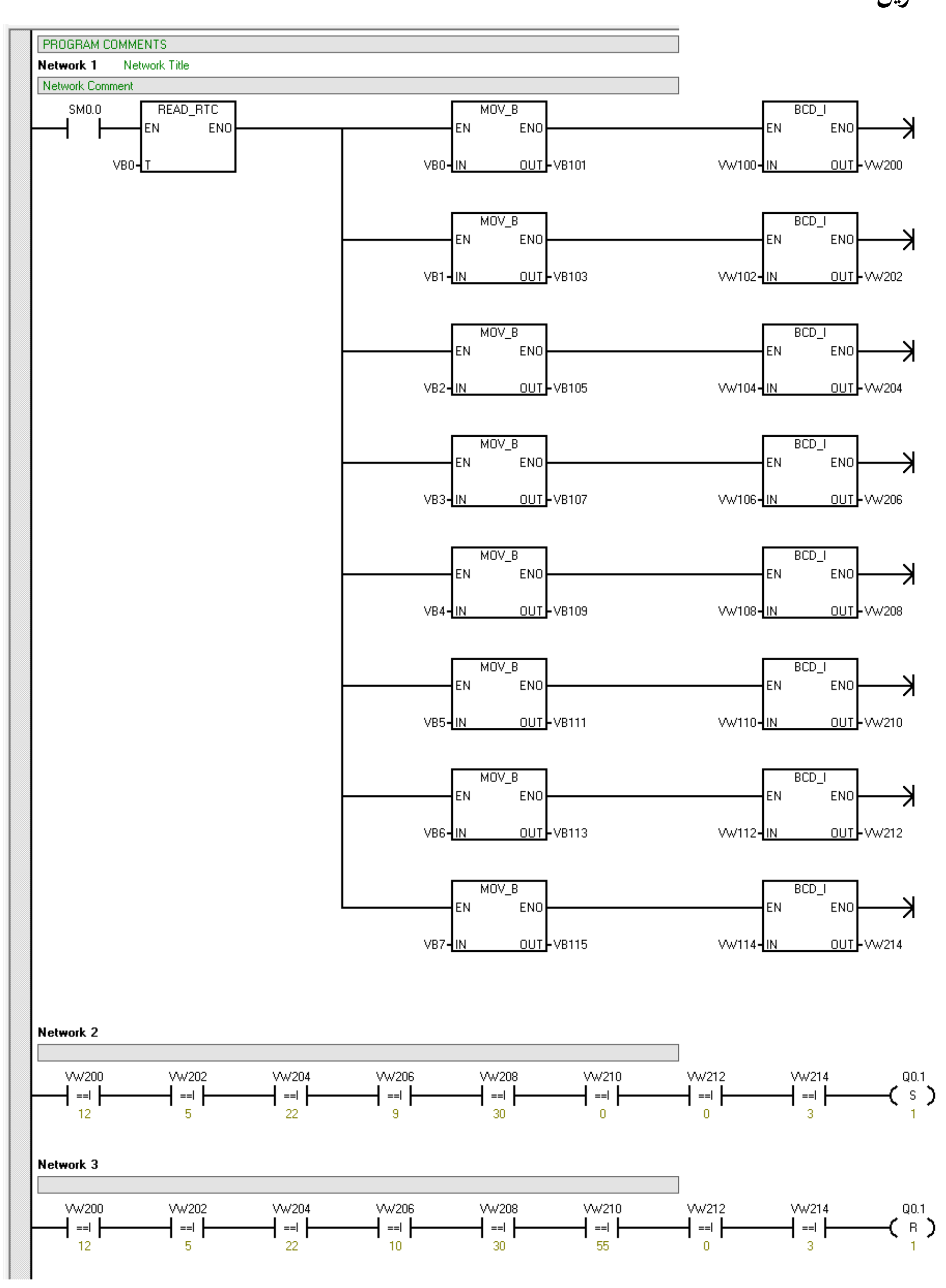
من المنطقي أنه قبل استخدام أو تنفيذ برنامج يعمل بالساعة والتاريخ أن يتم أولاً ضبطهما لذالك يستخدم أمر Read PC لضبط الساعة بحيث يتم ضبط الساعة بالاعتماد على ساعة الكمبيوتر.



ملاحظة:

قبل ضبط الساعة والتاريخ بوحدة البرمجة يجب أولاً ضبط الساعة والتاريخ الخاصين بالكمبيوتر.

التمرين


PROGRAM COMMENTS
Network 1
Network Title
Network Comment
SM0.0
READ_RTC
EN
ENO
VB0
T
MOV_B
IN
OUT
VB0
VB101
VB1
VB103
VB2
VB105
VB3
VB107
VB4
VB109
VB5
VB111
VB6
VB113
VB7
VB115
BCD_I
VW100
VW200
VW102
VW202
VW104
VW204
VW106
VW206
VW108
VW208
VW110
VW210
VW112
VW212
VW114
VW214
Network 2
VW200 ==I 12
VW202 ==I 5
VW204 ==I 22
VW206 ==I 9
VW208 ==I 30
VW210 ==I 0
VW212 ==I 0
VW214 ==I 3
Q0.1 S 1
Network 3
VW200 ==I 12
VW202 ==I 5
VW204 ==I 22
VW206 ==I 10
VW208 ==I 30
VW210 ==I 55
VW212 ==I 0
VW214 ==I 3
Q0.1 R 1

الشرح

الفرع الأول.

تم استخدام نقطة الريليه الخاص SM0.0 لكى تعمل الساعة بطريقة مستمرة و قد تم اختيار الـ VB0 لكى يستخدم كذاكرة محددة لتخزين الساعة والتاريخ بالترتيب التالى:

م	الذاكرة	الاستخدام	توضيح
١	VB0	لتخزين السنة	سنة ٢٠١٢ = ١٢
٢	VB1	لتخزين الشهر	شهر مايو = ٥
٣	VB2	لتخزين اليوم	يوم ١٨ فى الشهر = ١٨
٤	VB3	لتخزين الساعة	الساعة ٩ مساء = ٢١
٥	VB4	لتخزين الدقيقة	نصف ساعة = ٣٠
٦	VB5	لتخزين الثانية	ربع دقيقة = ١٥
٧	VB6	لتخزين اللحظة	لا تستخدم لذالك يكتب "صفر"
٨	VB7	لتخزين يوم فى الأسبوع	الأحد = ١, الاثنين = ٢ , الثلاثاء = ٣ وهكذا

نظراً إلى أن القيم المستخدمة مع الساعة والتاريخ لا تعمل كما نريد أى أنها تظهر بطريقة مختلفة فهى تظهر بلغة الـ BCD لذالك نقوم بالتحويل من الـ BCD إلى الـ Integer ولكن يجب أولاً تغير حجم الذاكرة من byte إلى word, للتوضيح أنظر الجدول التالى.

م	ماذا لدينا	ماذا نريد	كيف
١	ذاكرة بحجم byte	ذاكرة بحجم word	باستخدام أمر Move
٢	أرقاك BCD	أرقام Integer	باستخدام محول BCD-I

يتم أولاً تغير حجم الذاكرة من Byte إلى Word باستخدام أمر MOV-B (أنظر الجزء الأول من كتاب برمجة التحكم المنطقى) ثم يتم تغير القيمة من BCD إلى Integer باستخدام أمر التحويل BCD-I (أنظر الجزء الثانى من كتاب برمجة التحكم المنطقى) لثمان مرات متتالية (السنة – الشهر – اليوم – الساعة – الدقيقة – الثانية – اللحظة – اليوم فى الأسبوع)

الفرع الثانى.

سوف يعمل الخرج Q0.1 فى تاريخ ٢٠١٢/٥/٢٢ الساعة ٩ و ٣٠ دقيقة الموافق يوم الثلاثاء

الفرع الثالث.

سوف يتوقف الخرج Q0.1 فى تاريخ ٢٠١٢/٥/٢٢ الساعة ١٠ و ٣٠ دقيقة و ٥٥ ثانية الموافق يوم الثلاثاء

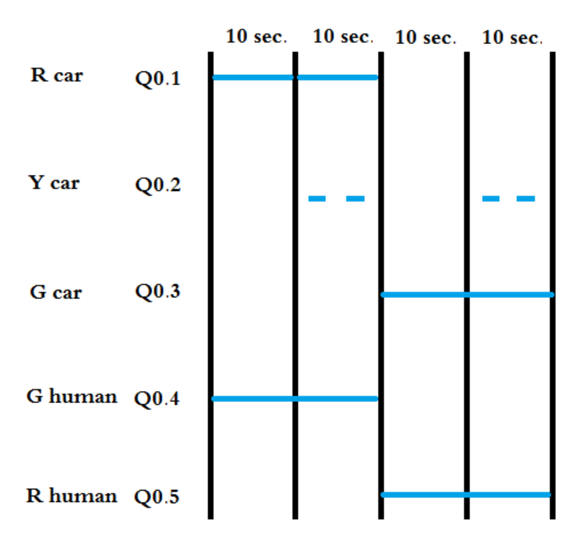
التمرين رقم ١٤

دائرة التحكم المنطقية لتنفيذ دائرة أشارة المرور كما بالشكل التالى.

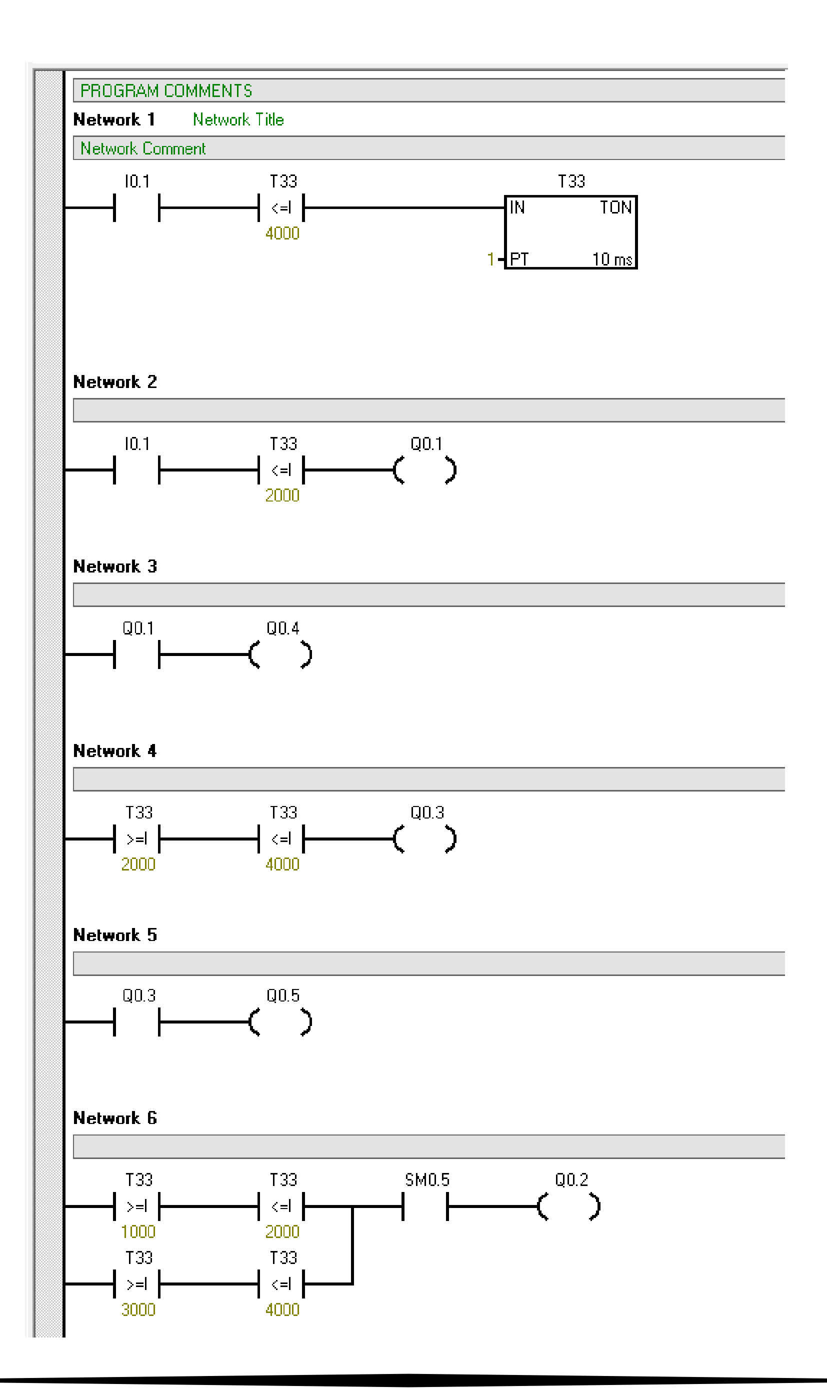
الجدول

م	النوع	الاسم	الاستخدام
١	input	I0.1	Selector
٢	output	Q0.1	Red "cars"
٣	output	Q0.2	Yellow "cars"
٤	output	Q0.3	Green "cars"
٥	output	Q0.4	Green "human"
٦	output	Q0.5	Red "human"

توضيح



التمرين


PROGRAM COMMENTS
Network 1
Network Title
Network Comment
I0.1
T33
<=I
4000
T33
IN
TON
1
PT
10 ms
Network 2
I0.1
T33
<=I
2000
Q0.1
Network 3
Q0.1
Q0.4
Network 4
T33
>=I
2000
T33
<=I
4000
Q0.3
Network 5
Q0.3
Q0.5
Network 6
T33
>=I
1000
T33
<=I
2000
SM0.5
Q0.2
T33
>=I
3000
T33
<=I
4000

الشرح

الفرع الأول.

بالضغط على I0.1 يعمل المؤقت الزمنى T33 لمدة ٤٠ ثانية ثم يبدأ من جديد.

الفرع الثانى.

تضيء الإشارة الحمراء الخاصة بالسيارات لمدة ٢٠ ثانية.

الفرع الثالث.

تضيء الإشارة الخضراء الخاصة بالمشاة مع الإشارة الحمراء الخاصة بالسيارات.

الفرع الرابع.

تضيء الإشارة الخضراء الخاصة بالسيارات لمدة ٢٠ ثانية أخرى.

الفرع الخامس.

تضيء الإشارة الحمراء الخاصة بالمشاة مع الإشارة الخضراء الخاصة بالسيارات.

الفرع الخامس.

تضيء الإشارة الصفراء بطريقة متقطعة كل نصف ثانية باستخدام SM0.5 لزمنين مختلفين يتكون كلاهما من ١٠ ثوانى.

التمرين رقم ١٥

دائرة التحكم المنطقية للتحكم بخزان كما بالشكل التالى.

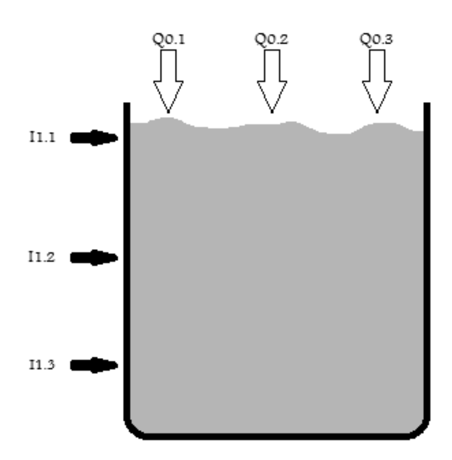
الجدول

م	النوع	الاسم	الاستخدام
١	input	I0.0	Stop
٢	input	I1.1	High Level
٣	input	I1.2	Operation Level
٤	input	I1.3	Low Level
٥	output	Q0.1	Pump-1
٦	output	Q0.2	Pump-2
	output	Q0.3	Pump-3

طريقة العمل.

تعمل الطلمبة الأولى Q0.1 عندما يقل مستوى السائل عن المستوى الثانى I1.2 للخزان بينما تعمل الطلمبة الثانية Q0.2 عندما يقل مستوى السائل عن المستوى الثالث I1.3 للخزان وعندما يبدأ المستوى فى الارتفاع لا تتوقف أى طلمبة حتى تصل إلى أعلى مستوى للخزان I1.1, أما بالنسبة للطلمبة الثالثة Q0.3 فهى تعمل بنفس طريقة عمل الطلمبة الثانية ولكن فى حالة الطوارئ فقط أى عندما يقل مستوى السائل من الأعلى I1.1 إلى الأسفل I1.3 فى أقل من نصف دقيقة

رسم للتوضيح.



الشرح

الفرع الأول.

تعمل الطلمبة الأولى عندما يقل مستوى السائل عن المستوى الثانى وتتوقف عندما يرتفع إلى المستوى الأول.

الفرع الثانى.

تعمل الطلمبة الثانية عندما يقل مستوى السائل عن المستوى الثالث وتتوقف عندما يرتفع إلى المستوى الأول.

الفرع الثالث.

يبدأ المؤقت الزمنى بالعمل عندما يقل مستوى السائل عن المستوى الأعلى.

الفرع الرابع.

تعمل الطلمبة الثالثة عندما يقل مستوى السائل من المستوى الأول إلى الثالث فى أقل من نصف دقيقة.

التمرين

PROGRAM COMMENTS

Network 1 Network Title

Network Comment

I1.2 / I1.1 / I0.0 Q0.1

Q0.1

Network 2

I1.3 / I1.1 / I0.0 Q0.2

Q0.2

Network 3

I1.1 / T32

IN TON

30000 PT 1 ms

Network 4

I1.3 / T32 / I1.1 / I0.0 Q0.3

Q0.3

التمرين رقم ١٦

دائرة التحكم المنطقية لمحرك عكس حركة يعمل تلقائياً بعد عودة التيار.

الجدول

م	النوع	الاسم	الاستخدام
١	n.c.	I0.0	Stop
٢	n.o.	I0.1	Start R
٣	n.o.	I0.2	Start L
٤	output	Q0.1	Motor R
٥	output	Q0.2	Motor L

الشرح

- الفرع الأول. بالضغط على I0.1 يعمل الريليه M0.1 ويتوقف بالضغط على I0.0.

- الفرع الثاني. يدور المحرك لليمين Q0.1 عندما يعمل الريليه M0.1.

- الفرع الثالث. بالضغط على I0.2 يعمل الريليه M0.2 ويتوقف بالضغط على I0.0.

- الفرع الرابع. يدور المحرك لليسار Q0.2 عندما يعمل الريليه M0.2.

التمرين

PROGRAM COMMENTS

Network 1 Network Title

Network Comment

I0.1 I0.0 M0.2 M0.1

M0.1

Network 2

M0.1 Q0.1

Network 3

I0.2 I0.0 M0.1 M0.2

M0.2

Network 4

M0.2 Q0.2

ملاحظة. تم استخدام الريليه لانه يعمل كذاكرة ويحتفظ بحالته حتى عند انقطاع مصدر الطاقة.

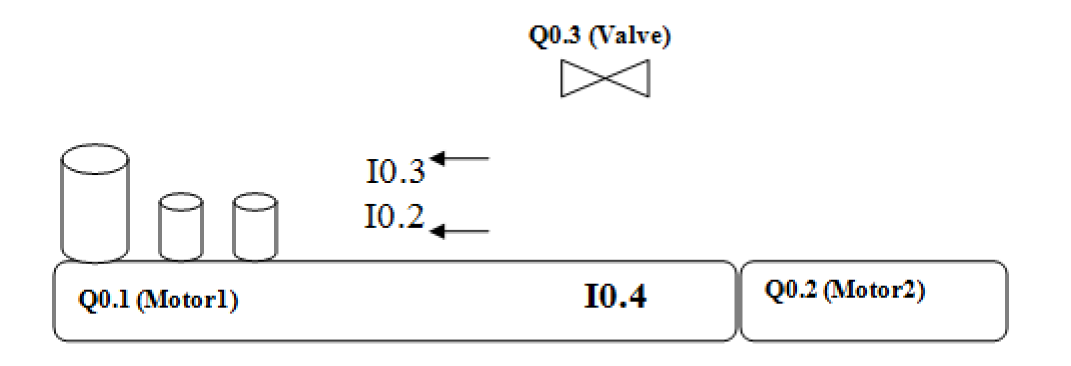
التمرين رقم ١٧

دائرة التحكم المنطقية لخط تعبئة زجاجات بمقاسات مختلفة.

الجدول

م	النوع	الاسم	الاستخدام
١	n.c.	I0.0	Stop
٢	n.o.	I0.1	Start
٣	n.o.	I0.2	Sensor High
٤	n.o.	I0.3	Sensor Low
٥	n.o.	I0.4	Sensor Position
٦	output	Q0.1	Line1
٧	output	Q0.2	Line2
٨	output	Q0.3	Valve

رسم توضيحي.



طريقة العمل.

بالضغط على مفتاح التشغيل I0.1 يعمل خط الإنتاج الأول Q0.1 والثانى Q0.2 فى نفس الاتجاه بحيث يحتوى خط الإنتاج الأول على زجاجات بمقاسات مختلفة بحيث يتم التميز بين المقاسين باستخدام الحساس I0.2. للزجاجات الصغيرة والحساس I0.3 للزجاجات الكبيرة ويتوقف خط الإنتاج الأول عندما تصل الزجاجة إلى الحساس I0.4 بحيث تكون الزجاجة فى الوضع الصحيح للمليء فيعمل الصمام Q0.3 أوتوماتيكياً لزمن محدد ثم يتوقف الصمام ويعمل خط الإنتاج الأول من جديد لكى تتكرر نفس العملية مع زجاجات أخرى.

إما بالنسبة للزمن المحدد لعمل الصمام فهو يعتمد على حجم الزجاجة وهو ما سوف يحدد بواسطة الحساسات.

يبقى خط الإنتاج الثانى Q0.2 فى العمل حتى يتم الضغط على مفتاح الإيقاف I0.0 ويتم استخدام هذا الأسلوب فى تقسيم خط الإنتاج إلى أقسام لتجنب أن تتوقف الزجاجة التى تم تعبئتها لمجرد أنه توجد زجاجة أخرى يتم تعبئها الآن.

الشرح

الفرع الأول.

بالضغط على **I0.1** يعمل خط الإنتاج الأول **Q0.1** ويتوقف بالضغط على **I0.0** أو عند الوصول إلى الحساس **I0.4**.

الفرع الثانى.

بالضغط على **I0.1** يعمل خط الإنتاج الثانى **Q0.2** ويتوقف بالضغط على **I0.0**.

الفرع الثالث.

عند وصول الزجاجة إلى المكان الخاص بالتعبئة يعمل الريليه لدورة واحدة ثم يتوقف.

التمرين

PROGRAM COMMENTS

Network 1 Network Title

Network Comment

I0.1 P I0.0 M0.4 / Q0.1

Q0.1

Q0.3 N

Network 2 Network Title

Network Comment

I0.1 I0.0 Q0.2

Q0.2

Network 3

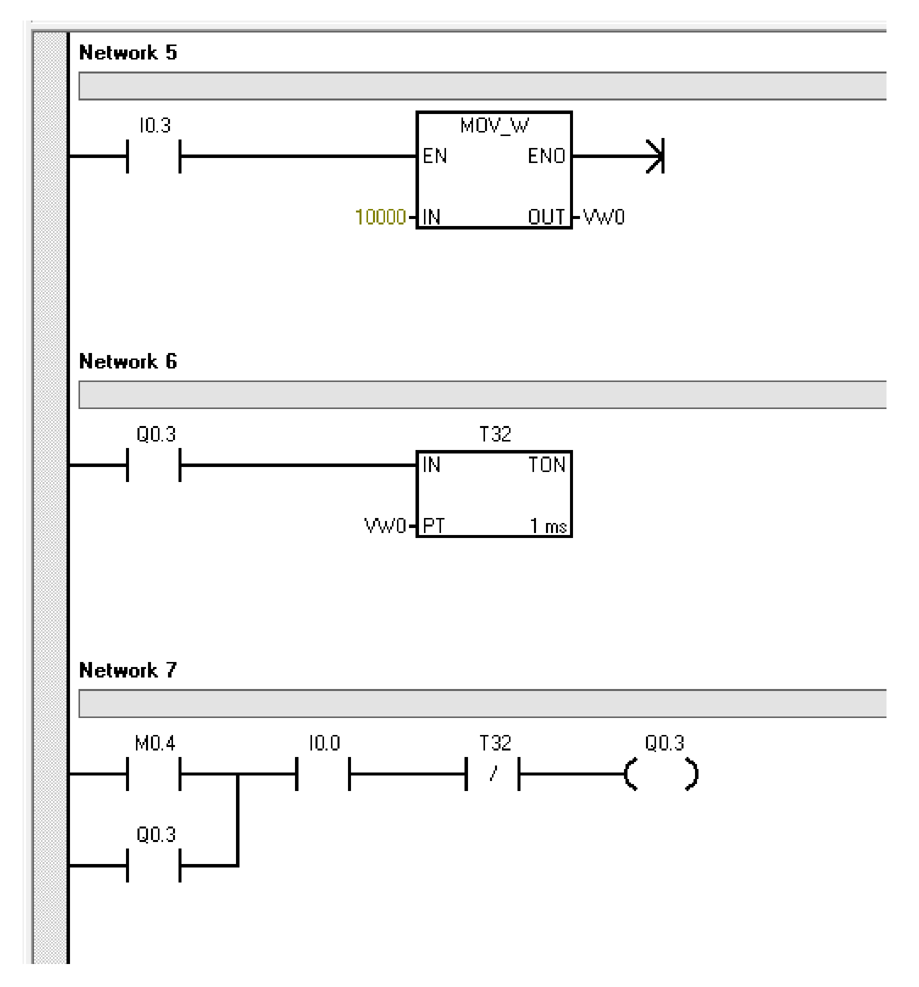
I0.4 P M0.4

Network 4

I0.2 MOV_W EN ENO

5000 IN OUT VW0

باقى التمرين.



الفرع الرابع.

إذا كانت الزجاجة صغيرة تصبح قيمة الذاكرة VW0 تساوى ٥ ثوانى.

الفرع الخامس.

إذا كانت الزجاجة كبيرة تصبح قيمة الذاكرة VW0 تساوى ١٠ ثوانى.

الفرع السادس.

يبدأ المؤقت الزمنى T32 فى العمل عندما يفتح الصمام Q0.3.

الفرع السابع.

يفتح الصمام Q0.3 عندما تصل الزجاجة إلى المكان المحدد للتعبئة.

التمرين رقم ١٨

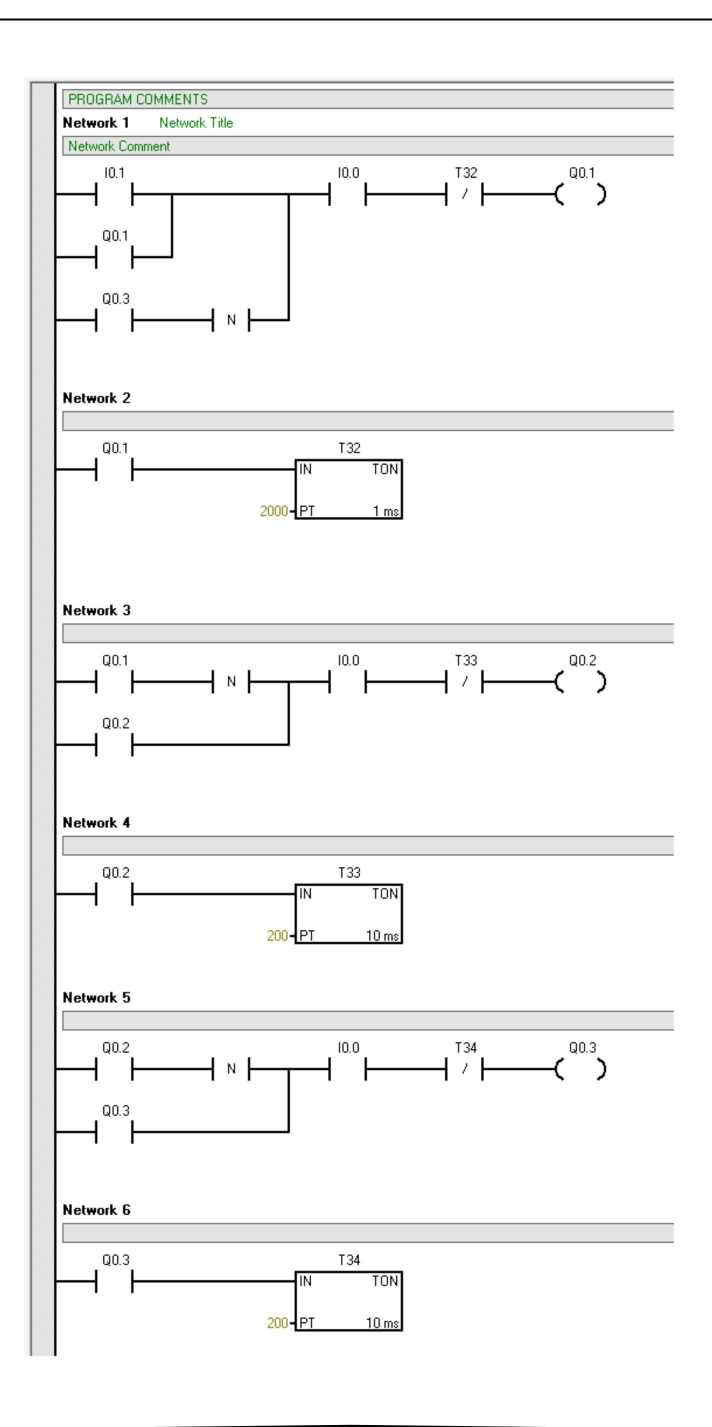
دائرة التحكم المنطقية لثلاث محركات تعمل بطريقة متتالية كما بالشكل التالى.

الجدول

م	النوع	الاسم	الاستخدام
١	n.c.	I0.0	Stop
٢	n.o.	I0.1	Start
٣	output	Q0.1	Motor1
٤	output	Q0.2	Motor2
٥	output	Q0.3	Motor3

رسم توضيحى.

التمرين



PROGRAM COMMENTS
Network 1
Network Title
Network Comment
I0.1
I0.0
T32
Q0.1
Q0.1
Q0.3
N
Network 2
Q0.1
T32
IN
TON
2000
PT
1 ms
Network 3
Q0.1
N
I0.0
T33
Q0.2
Q0.2
Network 4
Q0.2
T33
IN
TON
200
PT
10 ms
Network 5
Q0.2
N
I0.0
T34
Q0.3
Q0.3
Network 6
Q0.3
T34
IN
TON
200
PT
10 ms

الشرح

- الفرع الأول. بالضغط على I0.1 يعمل المحرك الأول Q0.1 ويتوقف بالضغط على I0.0 أو بعد مرور زمن محدد من بداية التشغيل بواسطة المؤقت الزمني T32.

- الفرع الثاني. يبدأ عمل المؤقت الزمني T32 عند عمل المحرك الأول.

- الفرع الثالث. بعد مرور زمن من عمل المحرك الأول ومن ثم يتوقف بواسطة المؤقت الزمني T32 فيعمل المحرك الثاني Q0.2 ويتوقف بالضغط على I0.0 أو بعد مرور زمن محدد من بداية التشغيل بواسطة المؤقت الزمني T33.

- الفرع الرابع. يبدأ عمل المؤقت الزمني T33 عند عمل المحرك الثاني.

- الفرع الخامس. بعد مرور زمن من عمل المحرك الثاني ومن ثم يتوقف بواسطة المؤقت الزمني T33 فيعمل المحرك الثالث Q0.3 ويتوقف بالضغط على I0.0 أو بعد مرور زمن محدد من بداية التشغيل بواسطة المؤقت الزمني T34.

- الفرع السادس. يبدأ عمل المؤقت الزمني T34 عند عمل المحرك الثالث.

الفهرس

الفهرس

الفهرس

الفهرس

الكتب التي صدرت عن معهد السالزيان الإيطالى "دون بوسكو"

🕮 **محركات, مولدات و محولات التيار المتردد**	وجيه جرجس
🕮 **دوائر التحكم الآلي** الجزء الأول	وجيه جرجس
🕮 **دوائر التحكم الآلي** الجزء الثاني	وجيه جرجس
🕮 **الغسالة الفول أوتوماتك** الجزء الأول	وجيه جرجس
🕮 **الغسالة الفول أوتوماتك** الجزء الثاني	وجيه جرجس
🕮 **الدوائر العملية للضغوط الهوائية و الكهروهوائية**	وجيه جرجس
🕮 **غسالة الأطباق**	وجيه جرجس
🕮 **زانوسى الموديلات القديمة ١٤-١٦-١٨بروجرام**	وجيه جرجس
🕮 **موديلات الغسالة كريازى**	وجيه جرجس
🕮 **الدوائر الكهربائية للتركيبات المنزلية**	نبيل رزق
🕮 **صيانة وإصلاح الأجهزة المنزلية**	نبيل رزق
🕮 **أفكار التكيف و التبريد للدوائر الميكانيكية**	إميل فتح الله
🕮 **أفكار التكيف و التبريد للدوائر الكهربائية**	إميل فتح الله
🕮 **أفكار التكيف و التبريد الخدمة والأعطال**	إميل فتح الله
🕮 **برمجة التحكم المنطقي .P.L.C** الجزء الأول	ريمون كمال
🕮 **برمجة التحكم المنطقي .P.L.C** الجزء الثاني	ريمون كمال
🕮 **برمجة التحكم المنطقي .P.L.C** أعطال - صيانة - تمارين	ريمون كمال